DOMAIN DECOMPOSITION METHODS FOR COUPLED STOKES-DARCY FLOWS

by

ChangQing Wang

Bachelor of Science in Mathematics,

Beijing Normal University, Beijing, China, 2010

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial

fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

ChangQing Wang

It was defended on

August 19th 2016

and approved by

Prof. Ivan Yotov, Dept. of Mathematics, University of Pittsburgh

Prof. William Layton, Dept. of Mathematics, University of Pittsburgh

Prof. Catalin Trenchea, Dept. of Mathematics, University of Pittsburgh

Prof. Paulo Zunino, Dept. of Mechanical Engineering & Materials Science, University of Pittsburgh

Dissertation Director: Prof. Ivan Yotov, Dept. of Mathematics, University of Pittsburgh

DOMAIN DECOMPOSITION METHODS FOR COUPLED STOKES-DARCY FLOWS

ChangQing Wang, PhD

University of Pittsburgh, 2016

This thesis studies the numerical methods for coupled Stokes-Darcy problem. It consists of three major parts: First, a non-overlapping domain decomposition method is presented for Stokes-Darcy problem by partitioning the computational domain into multiple subdomains, upon which families of coupled local problems of lower complexity are formulated. The coupling is based on appropriate interface matching conditions. The global problem is reduced to an interface problem by eliminating the interior subdomain variables, which can be solved by an iterative procedure. FETI approach is used for floating Stokes subdomains. The condition number of the resulting algebraic system is analyzed and numerical tests on matching grids verifying the theoretical estimates are provided. Second, a multiscale flux basis algorithm is developed based on the domain decomposition with multiscale mortar mixed finite element method. The algorithm involves precomputing a multiscale flux basis, which consists of the flux (or velocity trace) response from each mortar degree of freedom. It is computed by each subdomain independently before the interface iteration begins. The subdomain solves required at each iteration are substituted by a linear combination of the multiscale basis. This may lead to a significant reduction in computational cost since the number of subdomain solves is fixed, depending only on the number of mortar degrees of freedom associated with a subdomain. Several numerical examples are carried out to demonstrate the efficiency of the multiscale flux basis implementation. Third, a multiscale flux basis implementation is presented for coupled Stokes-Darcy flows with stochastic permeability, with its log represented as a sum of local Karhunen-Loève expansions. The problem

is approximated by stochastic collocation on either a tensor product or a sparse grid, coupled with multiscale mortar mixed finite element method using non-overlapping domain decomposition for the spatial discretization. Two algorithms based on deterministic or stochastic multiscale flux basis are introduced. Some numerical tests are presented to illustrate the performances of these algorithms, with the stochastic multiscale flux basis showing a great advantage in computational cost among all.

Keywords: non-overlapping domain decomposition, Stokes-Darcy flow, mortar finite element, mixed finite element, FETI method, multiscale flux basis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Ivan Yotov. He has always been patient in guiding me doing the research and helping me through all the troubles and difficulties. This work would not have been possible without his aids, his insights and considerate planning. I have learned a lot from the meetings with him during the last five years, especially on how to tackle the bigger issues by resolving them through all the fundamental and smaller pieces.

I would also like to thank professor William Layton, Catalin Trenchea and Paolo Zunino for serving in the thesis committee and providing me valuable suggestions on my research. I would like to thank all the helps from Ben Ganis, Pu Song, Eldar Khattatov and Ilona Ambartsumyan on the projects we work together.

Finally, a thank you to Yangyang, my fiancé, who cares about me and stays by my side every hour and moment over the last 5 years. It was like a dream come true that we came to US together after college, supported each other through all the ups and downs, and both earned our PhD degrees in the same year. I could not have made this accomplishment without you and our family.

1.0 INTRODUCTION

1.1 MODEL PROBLEM AND COUPLED STOKE-DARCY FLOW

Partial differential equations modeling various physical phenomena that arise in many fields of science and engineering often need to be coupled, since there are different processes taking place in different parts of the problem domain. Such problems consist of multiple models in different regions coupled through interface conditions. Even though the theories and numerical analysis of each model may be well developed, the coupled problem would still lead to various mathematical difficulties, such as handling different orders of equations in different regions, establishing the proper interface conditions and so on [40]. In this thesis we are studying a coupled Stokes-Darcy problem for fluid flow in porous media. Coupling Stokes and Darcy equations is an interesting topic because of its broad scope of possible applications: surface and subsurface water interaction, blood circulation, fuel cells, and filtration problems among others, where Stokes equation describes the motion of incompressible fluids and Darcy equation describes the infiltration process. We apply the Beavers-Joseph-Saffman slip with friction condition [7, 54] on the Stokes-Darcy interface. In some other literatures [13, 12], a more complicated Beaver-Joseph condition is used. Furthermore, the Stokes-Darcy flow can be coupled with a transport equation [59], which can be used, for example, to estimate the risk of groundwater contamination from chemicals discharged in rivers or lakes. In-depth understanding of the problem requires both modeling process and numerical study. In this thesis we will introduce a parallel algorithm to solve for the numerical solutions of the deterministic Stokes-Darcy model, which will also be extended to allow stochastic permeability in Chapter 4.

The problem model is given as follows. Assuming $d = 2, 3$, let $\Omega_S \subset \mathbb{R}^d$ be the fluid region

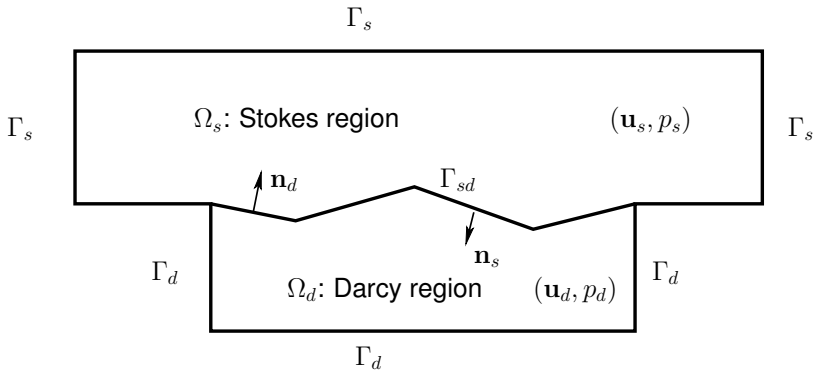

Figure 1.1: Stokes-Darcy model.

governed by the Stokes equations, with outside boundary Γ_S and outward unit normal $\mathbf{n}_S$. Let $\Omega_D \subset \mathbb{R}^d$ be the porous media region governed by Darcy's law, with outside boundary Γ_D and outward unit normal $\mathbf{n}_D$. Each region is a union of possibly disjoint subregions. Let $\Omega = \Omega_S \cup \Omega_D$ be the entire simulation domain and let $\Gamma_{SD} = \partial\Omega_S \cap \partial\Omega_D$ be the Stokes-Darcy interface (See Figure 1.1). The velocity and pressure in Ω_S and Ω_D are denoted by $\mathbf{u}_S$, p_S and $\mathbf{u}_D$, p_D, respectively. Let μ_S and μ_D be the viscosity coefficients in the Stokes and Darcy regions, and let $\mathbf{K}$ be the permeability tensor, assumed to be symmetric and uniformly positive definite. The deformation rate tensor $\mathbf{D}$ and the stress tensor $\mathbf{T}$ of the Stokes flow are denoted by

$$\mathbf{D}(\mathbf{u}_S) := \frac{1}{2}(\nabla \mathbf{u}_S + (\nabla \mathbf{u}_S)^T), \quad \mathbf{T}(\mathbf{u}_S, p_S) := -p_S \mathbf{I} + 2\mu_S \mathbf{D}(\mathbf{u}_S).$$

The Stokes flow model with no-slip boundary condition and body force $\mathbf{f}_S$ is:

$$-\nabla \cdot \mathbf{T}(\mathbf{u}_S, p_S) \equiv -2\mu_S \nabla \cdot \mathbf{D}(\mathbf{u}_S) + \nabla p_S = \mathbf{f}_S \quad \text{in } \Omega_S, \tag{1.1}$$

$$\nabla \cdot \mathbf{u}_S = 0 \quad \text{in } \Omega_S, \tag{1.2}$$

$$\mathbf{u}_S = \mathbf{0} \quad \text{on } \Gamma_S, \tag{1.3}$$

where (1.1) describes the conservation of momentum and (1.2) describes the conservation of mass. The Darcy flow model with no-flow boundary condition, gravity force $\mathbf{f}_D$ and external source q_D is:

$$\mu_D \mathbf{K}^{-1}\mathbf{u}_D + \nabla p_D = \mathbf{f}_D \quad \text{in } \Omega_D, \tag{1.4}$$

$$\nabla \cdot \mathbf{u}_D = q_D \quad \text{in } \Omega_D, \tag{1.5}$$

$$\mathbf{u}_D \cdot \mathbf{n}_D = 0 \quad \text{on } \Gamma_D, \tag{1.6}$$

where (1.4) is the Darcy's law and (1.5) describes the conservation of mass. We assume that the source q_D satisfies the solvability condition $\int_{\Omega_D} q_D = 0$. The interface conditions on Γ_{SD} are:

$$\mathbf{u}_S \cdot \mathbf{n}_S + \mathbf{u}_D \cdot \mathbf{n}_D = 0 \quad \text{on } \Gamma_{SD}, \tag{1.7}$$

$$-(\mathbf{T}(\mathbf{u}_S, p_S)\mathbf{n}_S) \cdot \mathbf{n}_S \equiv p_S - 2\mu_S(\mathbf{D}(\mathbf{u}_S)\mathbf{n}_S) \cdot \mathbf{n}_S = p_D \quad \text{on } \Gamma_{SD}, \tag{1.8}$$

$$-(\mathbf{T}(\mathbf{u}_S, p_S)\mathbf{n}_S) \cdot \boldsymbol{\tau}_{SD}^l \equiv -2\mu_S(\mathbf{D}(\mathbf{u}_S)\mathbf{n}_S) \cdot \boldsymbol{\tau}_{SD}^l = \frac{\mu_S \alpha}{\sqrt{K_l}} \mathbf{u}_S \cdot \boldsymbol{\tau}_{SD}^l, \quad 1 \le l \le d-1 \quad \text{on } \Gamma_{SD}. \tag{1.9}$$

Conditions (1.7) and (1.8) incorporate continuity of flux and normal stress, respectively. Condition (1.9) represents the Beaver-Joseph-Saffman slip with friction condition [7, 54], where $K_l = (\mathbf{K}\boldsymbol{\tau}_{SD}^l) \cdot \boldsymbol{\tau}_{SD}^l$, $\{\boldsymbol{\tau}_{SD}^l\}_{l=1}^{d-1}$ is an orthogonal system of unit tangent vectors on Γ_{SD}, and the constant $\alpha \ge 0$ is determined experimentally.

1.2 INTRODUCTION TO METHODOLOGY

A variety of numerical methods exist for the coupled Stokes-Darcy problem. Among them there are two major types, one is to solve the coupled problem in a unified approach using the same finite element space [5, 39, 49]; the other is to decouple the problem at first and then apply appropriate local solver, such as domain decomposition methods[40, 19, 46, 53, 27], two grid methods [47], etc.

Domain decomposition methods [41, 51, 57], naturally lead to designing parallel algorithms and allow different numerical schemes within different subdomains to be employed, which makes them very attractive for multiphysics problems. Another advantage is the possibility to reuse existing computer code libraries for the subdomain problems.

In this thesis we utilize the multiscale mortar mixed finite element method (MMMFEM) with non-overlapping domain decomposition (DD) [40, 34, 55, 58]. In this method, the computational domain is decomposed into several subdomains of either Stokes or Darcy type. A non-overlapping domain decomposition for the Stokes-Darcy problem with many subdomains was first developed and analyzed in [58] following the approach from [35]. Earlier works in two-subdomain case include [18, 20, 21, 36, 28]. In [58], by eliminating the subdomain unknowns, the fine scale global problem is reduced to an interface problem that can be solved by an iterative method. The action of the interface operator requires solving in parallel either local Stokes or Darcy subdomain problems of lower complexity. Each subdomain is discretized on a local fine mesh, allowing for non-matching grids across the subdomain interfaces. This capability is helpful in practice, since it allows for describing complex geometries as a union of simpler locally discretized subdomains, as well as resolving internal features such as geological layers and faults.

Because of the possibility of using multiple subdomains, we need to account for interfaces of the three types: Stokes-Darcy, Darcy-Darcy and Stokes-Stokes. On the Stokes-Darcy interfaces the conditions are continuity of the normal velocity and normal stress, as well as the Beavers-Joseph-Saffman [7, 54] condition for the tangential Stokes velocity [40]. On the Stokes-Stokes interfaces the velocity vector and the normal stress vector are continuous, while on the Darcy-Darcy interfaces the normal velocity and pressure are continuous. For simplicity of notation, we restrict the presentation to two connected regions, one Stokes and one Darcy, each one subdivided into multiple subdomains. However, the formulation and the analysis naturally extend to the case when the Stokes and Darcy regions consist of multiple disconnected components.

A mortar finite element space is introduced on the interfaces, which serves as a Lagrange multiplier to impose weakly appropriate continuity conditions: normal velocity and normal stress on Stokes-Darcy interfaces, normal velocity and pressure on Darcy-Darcy interfaces,

and velocity vector and normal stress vector on Stokes-Stokes interfaces. The numerical analysis of the method is carried out in [34] on fairly general grid configurations, allowing for multiscale approximations with coarse scale H mortar spaces and fine scale h subdomain discretizations. The work in [34] extends earlier works that consider a single Stokes-Darcy interface and employ a mortar space defined as the normal trace of the Darcy velocity space [40, 53, 27, 9].

In our algorithm, computing the action of the interface operator requires solving Stokes subdomain problems of Neumann or Neumann-Robin type and Darcy subdomain problems of Dirichlet type. As a result, the Stokes subdomain problems can be singular. We employ an approach based on the finite element tearing and interconnecting (FETI) method [24, 57], by projecting the iterates into a space orthogonal to the kernel of the subdomain operators via solving an auxiliary coarse space problem to ensure that the local Stokes problems are solvable.

In [58], we established that for matching grids case (i.e. $H = h$) the interface operator S_h (h is the mesh size) is symmetric and positive definite and showed that the different interface types have different effect on its condition number. More precisely, $\text{cond}(S_h) = O(h^{-1}A^{-1})$ if there are no Stokes-Stokes interfaces present, and $\text{cond}(S_h) = O(h^{-2}A^{-1})$ otherwise, where A is the characteristic subdomain diameter. Furthermore, very small values of the Darcy permeability dominate the discretization effect, in which case we have $\text{cond}(S_h) = O(k^{-1}A^{-1})$ or $\text{cond}(S_h) = O(k^{-1}h^{-1}A^{-1})$ in the cases with or without Stokes-Stokes interfaces, respectively, where k is the characteristic permeability value. We note that our formulation is suitable for the application of optimal interface preconditioners, such as balancing [16, 44, 45, 50, 28], which should eliminate the dependence on h and A in the condition number, see [28] for the two-subdomain case.

The main computational cost of the above method comes from the subdomain solves required in every interface iteration. Increasing the number of subdomains and refining the grids both lead to an increase in the number of iterations and the number of subdomain solves. We develop an alternative implementation of the MMMFEM for Stokes-Darcy flows that is based on precomputing a multiscale flux basis, which can reduce the computational cost significantly. Our approach extends the multiscale flux basis implementation of the

MMMFEM for Darcy flow developed in [31]. The MMMFEM was first developed in [4] for Darcy problems. It is an alternative to other existing multiscale methods such as the variational multiscale method [38, 3] and the multiscale finite element method [37, 15]. Methods involving enriched multiscale basis for high-contrast problems using local spectral information have been developed in [23, 22, 26]. All three methods require solving relatively small fine scale subdomain problems that are only coupled on the coarse scale through a reduced number of degrees of freedom. The mortar multiscale approach provides the extra flexibility to adaptively refine the mortar grids based on a posteriori error estimation in order to improve the global accuracy [4]. The variational multiscale method and multiscale finite elements both compute a multiscale basis by solving a fixed number of local fine scale problems with boundary conditions or a source term corresponding to the coarse scale degrees of freedom. This basis is then used to solve the coarse scale problem. The multiscale flux basis implementation of the MMMFEM for Stokes-Darcy flows to be introduced in this thesis [30] provides a similar computational structure. The method yields the same solution as the original MMMFEM implementation but can be much more computationally efficient.

A multiscale flux basis consists of the flux (or velocity trace) response from each mortar degree of freedom, which is computed by each subdomain independently before the interface iteration begins. Then the subdomain solves during the interface iteration can be replaced by linear combinations of the multiscale flux basis. This implementation has a number of fine scale subdomain solves that is independent of the number of interface iterations. It reduces the computational cost if there are more iteration steps than number of mortar degree of freedoms per subdomain. In addition, when performing studies where the same input data is used repeatedly in different situations, the multiscale flux basis can be computed once and stored to disk in an offline step, so it can be reused across different simulations. Even though it was first introduced for deterministic case, a typical extension for this approach is the stochastic Stokes-Darcy flow problem [2], where the permeability in the Darcy region is given as a stochastic parameter presented by a sum of the local Karhunen-Loève (KL) expansion [33]. We will discuss in Chapter 4 about the stochastic multiscale flux basis and its application to uncertainty quantification problems.

1.3 THESIS OUTLINE

The rest of the thesis is organized as follows: In Chapter 2 we present the non-overlapping domain decomposition formulation for the coupled Stokes-Darcy model, derived its variational form, as well as the MMMFEM discretization, and the reduction to a mortar interface problem. Details about implementing FETI method for floating Stokes subdomains are also included. Some numerical experiments are presented for the matching mesh (non-mortar) case. In Chapter 3, the multiscale flux basis algorithm is developed for Stokes-Darcy flow and its combination with balancing preconditioner. In Chapter 4 we present one traditional and two multiscale flux basis implementations for coupled Stokes and Darcy flows with stochastic permeability.

2.0 DOMAIN DECOMPOSITION AND MORTAR MIXED FINITE ELEMENT DISCRETIZATION

We will introduce the domain decomposition method and the mortar mixed finite element discretization for the Stokes-Darcy model (1.1) - (1.9). In the following discussions, the L^2-inner product and norm of scalar and vector valued functions in domain $G \subset \mathbb{R}^d$ are denoted by $(\cdot,\cdot)_G$ and $\|\cdot\|_G$, respectively. To simplify the notation, we omit the subscript G if $G = \Omega$. For a section of the interface or the domain boundary $S \subset \mathbb{R}^{d-1}$, the L^2-inner product (or duality pairing) and norm are denoted by $\langle\cdot,\cdot\rangle_S$ and $\|\cdot\|_S$, respectively.

2.1 DOMAIN DECOMPOSITION METHOD ON STOKES-DARCY FLOW

The domain Ω is decomposed into N non-overlapping subdomians Ω_i, $i = 1, \ldots, N$, where

$$\Omega_S = \cup_{i=1}^{N_S}\Omega_i, \quad \Omega_D = \cup_{i=N_S+1}^{N}\Omega_i, \quad N = N_S + N_D.$$

For $1 \le i < j \le N$, we define $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ as the interface between any two subdomains, which can be of zero measure if they are not adjacent. Define $\Gamma_i = \cup_j \Gamma_{ij}$ as the union of interfaces of subdomain Ω_i and $\Gamma = \cup_{i,j}\Gamma_{i,j}$ as the union of all interfaces. Let Γ_{SS} be the set of Stokes-Stokes interfaces and let Γ_{DD} be the set of Darcy-Darcy interfaces. The following interface conditions are imposed:

$$[\mathbf{u}_D \cdot \mathbf{n}] = 0,\ [p_D] = 0 \text{ on } \Gamma_{DD}, \quad [\mathbf{u}_S] = \mathbf{0},\ [\mathbf{T}(\mathbf{u}_S, p_S)\mathbf{n}] = \mathbf{0} \text{ on } \Gamma_{SS}, \tag{2.1}$$

where $[\ \cdot\ ]$ denotes the jump on the interface. In particular, on Γ_{ij}, $[p] = (p_i - p_j)|_{\Gamma_{ij}}$, $[\mathbf{u}\cdot\mathbf{n}] = \mathbf{u}_i \cdot \mathbf{n}_i + \mathbf{u}_j \cdot \mathbf{n}_j$, with the notation $\mathbf{u}_i := \mathbf{u}|_{\Omega_i}$, $p_i := p|_{\Omega_i}$, $\mathbf{n}_i$ being the outward

unit normal vector to $\partial\Omega_i$. Also, denote $\mathbf{f}_i = \mathbf{f}_S|_{\Omega_i}$ for $1 \le i \le N_S$ and $\mathbf{f}_i = \mathbf{f}_D|_{\Omega_i}$ for $N_S + 1 \le i \le N$.

Following the variational formulation derived in [34], we define the velocity and pressure spaces

$$V^S = \{\mathbf{v}_S \in (L^2(\Omega_S))^d : \mathbf{v}_S|_{\Omega_i} \in (H^1(\Omega_i))^d, 1 \le i \le N_S, \mathbf{v}_S = 0 \ \text{ on } \Gamma_S\}, \ W^S = L^2(\Omega_S)$$

in the Stokes region Ω_S, and

$$V^D = \{\mathbf{v}_D \in (L^2(\Omega_D))^d : \mathbf{v}_D|_{\Omega_i} \in H(\mathrm{div}; \Omega_i), N_s + 1 \le i \le N, \mathbf{v}_D \cdot \mathbf{n}_D = 0 \ \text{ on } \Gamma_D\},$$
$$W^D = L^2(\Omega_D)$$

in the Darcy region Ω_D, where

$$H(\mathrm{div}; \Omega_i) = \{\mathbf{v} \in (L^2(\Omega_i))^d : \nabla \cdot \mathbf{v} \in L^2(\Omega_i)\}.$$

The velocity and pressure spaces on the whole domain are given by $V = V^S \times V^D$ and

$$W = \left\{ w = (w_S, w_D) \in W^S \times W^D : \int_\Omega w = 0 \right\}.$$

To impose the continuity conditions on the interfaces we define the Lagrange multiplier space

$$\Lambda = \Lambda^{SD} \times \Lambda^{DD} \times \Lambda^{SS}, \ \Lambda^{DD} = (V^D \cdot \mathbf{n}|_{\Gamma_{DD}})', \ \Lambda^{SD} = (V^D \cdot \mathbf{n}|_{\Gamma_{SD}})', \ \Lambda^{SS} = (V^S|_{\Gamma_{SS}})'.$$

The variational formulation of the coupled problems (1.1) – (1.9) is: find $(\mathbf{u}, p, \lambda) \in V \times W \times \Lambda$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + b_I(\mathbf{v}, \lambda) = (\mathbf{f}, \mathbf{v}), \quad \forall\, \mathbf{v} \in V, \tag{2.2}$$

$$b(\mathbf{u}, w) = -(q_D, w)_{\Omega_D}, \quad \forall\, w \in W, \tag{2.3}$$

$$b_I(\mathbf{u}, \mu) = 0, \quad \forall\, \mu \in \Lambda, \tag{2.4}$$

where

$$a_i(\mathbf{u},\mathbf{v}) = 2\,\mu_S(\mathbf{D}(\mathbf{u}_i),\mathbf{D}(\mathbf{v}_i))_{\Omega_i} + \sum_{l=1}^{d-1}\left\langle \frac{\mu_S\alpha}{\sqrt{K_l}}(\mathbf{u}_i\cdot\boldsymbol{\tau}_l)(\mathbf{v}_i\cdot\boldsymbol{\tau}_l)\right\rangle_{\partial\Omega_i\cap\Gamma_{SD}},$$
$$1\le i\le N_S,\quad \forall\,(\mathbf{u},\mathbf{v})\in V^S\times V^S,$$

$$a_i(\mathbf{u},\mathbf{v}) = \mu_D(\mathbf{K}^{-1}\mathbf{u}_i,\mathbf{v}_i)_{\Omega_i},\quad N_S+1\le i\le N,\quad \forall\,(\mathbf{u},\mathbf{v})\in V^D\times V^D,$$

$$b_i(\mathbf{v},w) = -(\nabla\cdot\mathbf{v}_i,w_i)_{\Omega_i},\quad 1\le i\le N,\quad \forall\,\mathbf{v}\in V,\quad \forall\,w\in W,$$

$$a(\mathbf{u},\mathbf{v}) = \sum_{i=1}^{N} a_i(\mathbf{u},\mathbf{v}),\quad b(\mathbf{v},w) = \sum_{i=1}^{N} b_i(\mathbf{v},w),$$

$$b_I(\mathbf{v},\mu) = \langle[\mathbf{v}],\mu\rangle_{\Gamma_{SS}} + \langle[\mathbf{v}\cdot\mathbf{n}],\mu\rangle_{\Gamma_{DD}} + \langle[\mathbf{v}\cdot\mathbf{n}],\mu\rangle_{\Gamma_{SD}},\quad \forall\,(\mathbf{v},\mu)\in V\times\Lambda.$$

The Lagrange multiplier λ has the physical meaning of normal stress vector on Γ_{SS} and pressure on $\Gamma_{DD}\cup\Gamma_{SD}$. Equation (2.4) is needed to weakly enforce the continuity conditions (1.7) and (2.1) on the different types of interfaces. The reader is referred to [34, 40] for proof of the existence and uniqueness of a solution to the variational formulation (2.2)–(2.4).

2.2 MORTAR MIXED FINITE ELEMENT DISCRETIZATION

For simplicity, we denote $V_i = V|_{\Omega_i}$ for $1\le i\le N$. In the MMMFEM [34, 58], each subdomain Ω_i is discretized with a d-dimensional shape regular finite element partition $\mathcal{T}_{h_i}$, where h_i is the maximal element diameter. For any adjacent subdomains Ω_i and Ω_j, the partitions $\mathcal{T}_{h_i}$ and $\mathcal{T}_{h_j}$ need not match on Γ_{ij}. In addition, a coarse $(d-1)$-dimensional quasi-uniform affine mesh $\mathcal{T}_{H_{ij}}$ is defined on the interface Γ_{ij} with maximal element size H_{ij}. Let $h = \max_{i=1}^{N} h_i$ and $H = \max_{i,j} H_{ij}$. In any Stokes subdomain Ω_i, $1\le i\le N_S$, let $V_{h,i}\times W_{h,i}\subset V_i\times W_i$ be a pair of finite element spaces satisfying the following discrete inf-sup condition for some constant $\beta_S>0$:

$$\inf_{0\neq w_{h,i}\in W_{h,i}}\ \sup_{0\neq\mathbf{v}_{h,i}\in V_{h,i}}\ \frac{(w_{h,i},\nabla\cdot\mathbf{v}_{h,i})_{\Omega_i}}{\|\mathbf{v}_{h,i}\|_{H^1(\Omega_i)}\ \|w_{h,i}\|_{L^2(\Omega_i)}}\ge\beta_S>0,\quad 1\le i\le N_S. \tag{2.5}$$

Some well-known examples of pairs satisfying (2.5) are the Taylor-Hood element [56], the MINI element [6], and the Bernardi-Raugel element [8]. In any Darcy subdomain Ω_i, N_S+

$1 \leq i \leq N$, let $V_{h,i} \times W_{h,i} \subset V_i \times W_i$ be a pair of mixed finite element spaces satisfying $\nabla \cdot V_{h,i} \subset W_{h,i}$ and the discrete inf-sup condition for some constant $\beta_D > 0$:

$$\inf_{0 \neq w_{h,i} \in W_{h,i}} \sup_{0 \neq \mathbf{v}_{h,i} \in V_{h,i}} \frac{(w_{h,i}, \nabla \cdot \mathbf{v}_{h,i})_{\Omega_i}}{\|\mathbf{v}_{h,i}\|_{H(\mathrm{div};\Omega_i)} \quad \|w_{h,i}\|_{L^2(\Omega_i)}} \geq \beta_D > 0, \quad N_S + 1 \leq i \leq N. \tag{2.6}$$

Well-known pairs that satisfy these conditions include the Raviart-Thomas spaces [52], the Brezzi-Douglas-Marini (BDM) spaces [11], and the Brezzi-Douglas-Duran-Fortin (BDDF) spaces [10]. On each interface Γ_{ij}, a mortar space $\Lambda_{H,ij} \subset L^2(\Gamma_{ij})$ if $\Gamma_{ij} \subset \Gamma_{SD} \cup \Gamma_{DD}$ or $\Lambda_{H,ij} \subset (L^2(\Gamma_{ij}))^d$ if $\Gamma_{ij} \subset \Gamma_{SS}$ is defined to weakly impose the continuity conditions for the discrete velocity across the non-matching grids. These mortar spaces consist of continuous or discontinuous piecewise polynomials of degree that may vary on the different types of interfaces. Globally, the finite element spaces are defined as

$$V_h = \bigoplus_{i=1}^{N} V_{h,i}, \quad W_h = \bigoplus_{i=1}^{N} W_{h,i}, \quad \Lambda_H = \bigoplus_{i,j} \Lambda_{H,ij}.$$

In the multiscale mortar mixed finite element discretization of (2.2)–(2.4) we seek $(\mathbf{u}_h, p_h, \lambda_H) \in V_h \times W_h \times \Lambda_H$, such that

$$a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) + b_I(\mathbf{v}_h, \lambda_H) = (\mathbf{f}, \mathbf{v}_h), \quad \forall\, \mathbf{v}_h \in V_h, \tag{2.7}$$

$$b(\mathbf{u}_h, w_h) = -(q_D, w_h)_{\Omega_D}, \quad \forall\, w_h \in W_h, \tag{2.8}$$

$$b_I(\mathbf{u}_h, \mu_H) = 0, \quad \forall\, \mu_H \in \Lambda_H. \tag{2.9}$$

The following convergence result for (2.7)–(2.9) has been shown in [34]:

Theorem 2.2.1. *Assume that the solution to (2.2)–(2.4) is sufficiently smooth, and let r_S and r_D be the polynomial degrees of the velocity spaces in Stokes and Darcy respectively, and let m_S, m_D, and m_{SD} be the polynomial degrees of the mortar spaces on Γ_{SS}, Γ_{DD}, and Γ_{SD}, respectively. Then there exists a positive constant C independent of h and H such that*

$$\|\mathbf{u} - \mathbf{u}_h\|_V + \|p - p_h\|_W \leq C(h^{r_S} + h^{r_D+1} + H^{m_S+1/2} + H^{m_D+1/2} + H^{m_{SD}+1/2}).$$

Here we also present a couple of results regarding the coercivity and continuity of $a_i(\cdot,\cdot)$ which is important for the well-posedness of the subdomain problem.

Recall that the kernel of the operator $\mathbf{D}$ consists of all rigid body motions

$$\mathcal{RB} = \begin{cases} \{\mathbf{a} + b[x_2, -x_1]^T\}, & d = 2, \\ \{\mathbf{a} + \mathbf{b} \times [x_1, x_2, x_3]^T\}, & d = 3, \end{cases}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. The space spans two translations and one rotation in $\mathbb{R}^2$, and three translations and three rotations in $\mathbb{R}^3$. Let

$$\ker\, a_i = \{\mathbf{v} \in V_i^S : a_i(\mathbf{v}, \mathbf{v}) = 0\}.$$

Since $\mathbf{K}$ is uniformly positive definite, we have ker $a_i = \mathbf{0}$ in the Darcy region. The following lemma describes the kernels of the subdomain bilinear forms in the Stokes region. We assume that the subdomain boundaries cannot intersect $\partial\Omega$ only along an edge in $\mathbb{R}^3$ and at a point in $\mathbb{R}^2$.

Lemma 2.2.2. *There are several possible cases for Stokes subdomains,* $1 \le i \le N_S$*:*

- *if* $\partial\Omega_i \subset \Gamma_{SS}$*, then* ker $a_i = \mathcal{RB}$*,*
- *if* $\partial\Omega_i \cap \partial\Omega \neq \varnothing$*, then* ker $a_i = \mathbf{0}$*,*
- *if* $\partial\Omega_i \cap \partial\Omega = \varnothing$ *and* $\partial\Omega_i$ *contains exactly one segment of* Γ_{SD}*, then in* $\mathbb{R}^3$ ker a_i *spans one translation and two rotations and in* $\mathbb{R}^2$ *it spans one translation and one rotation that are orthogonal to the tangent vectors on the* Γ_{SD} *segment,*
- *if* $\partial\Omega_i \cap \partial\Omega = \varnothing$ *and* $\partial\Omega_i$ *contains exactly two (non-connected) parallel segments of* Γ_{SD}*, then* ker a_i *spans one translation that is orthogonal to the tangent vectors on the segments,*
- *if* $\partial\Omega_i \cap \partial\Omega = \varnothing$ *and* $\partial\Omega_i$ *contains exactly two non-parallel segments of* Γ_{SD}*, then* ker a_i *spans one rotation that is orthogonal to the tangent vectors on the segments,*
- *if* $\partial\Omega_i$ *contains more that two segments of* Γ_{SD}*, then* ker $a_i = \mathbf{0}$*.*

Also the following lemma proved in [58] gives the coercivity and continuity of $a_i(\cdot,\cdot)$:

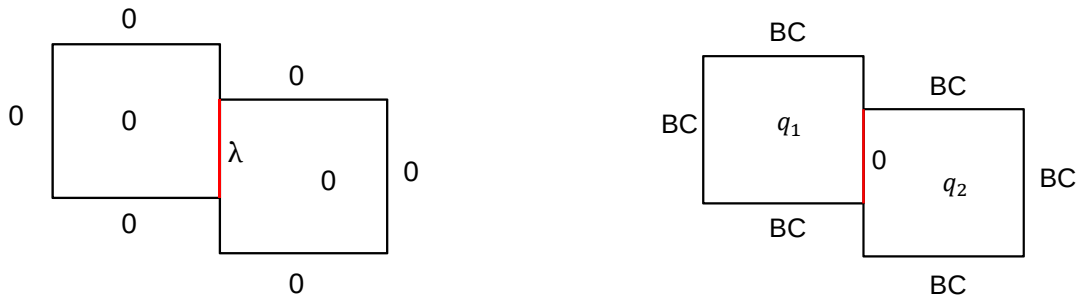

Figure 2.1: Local problems for $(\mathbf{u}^*, p^*)$ (left) and $(\overline{\mathbf{u}}, \overline{p})$ (right)

Lemma 2.2.3. *There exist positive constants C_1, C_2, C_3, and C_4 such that*

$$\begin{aligned} &\textit{for} \;\; 1 \le i \le N_S, \\ &C_1\|\mathbf{v}_i\|^2_{1,\Omega_i} \le a_i(\mathbf{v}_i, \mathbf{v}_i) \le C_2\|\mathbf{v}_i\|^2_{1,\Omega_i}, \quad \forall \mathbf{v} \in V_{h,i}/\ker\, a_i, \end{aligned} \tag{2.10}$$

$$\begin{aligned} &\textit{for} \;\; N_S + 1 \le i \le N, \\ &C_3\|\mathbf{v}_i\|^2_{\Omega_i} \le a_i(\mathbf{v}_i, \mathbf{v}_i) \le C_4\|\mathbf{v}_i\|^2_{\Omega_i}, \quad \forall \mathbf{v} \in V_{h,i}. \end{aligned} \tag{2.11}$$

2.3 REDUCTION TO AN INTERFACE PROBLEM

Following the algorithm in [58], we reduce the discretized problem (2.7)−(2.9) to an interface problem, which can be solved using a Krylov iterative solver. The original problem is split into two families of local problems on each Ω_i, one with zero source, zero outside boundary conditions, and specified interface value; the other with zero interface value, specified source, and specified outside boundary conditions. Correspondingly, the solution to (2.7) − (2.9) is decomposed into $\mathbf{u}_h = \mathbf{u}^*_h + \overline{\mathbf{u}}_h$, $p_h = p^*_h + \overline{p}_h$, see Figure 2.1. On each Stokes subdomain, let $\lambda = (\lambda_n, \lambda_\tau)$, where λ_n and $\lambda_\tau = (\lambda^1_\tau, \ldots, \lambda^{d-1}_\tau)$ represent the normal stress and tangential stress on Γ_{SS}, respectively, Consider the set of Stokes subdomain problems with specified normal and tangential stress on the interfaces: find $(\mathbf{u}^*_{h,i}(\lambda), p^*_{h,i}(\lambda)) \in V_{h,i}/\ker\, a_i \times W_{h,i}$,

$1 \le i \le N_S$, such that

$$a_i(\mathbf{u}^*_{h,i}(\lambda), \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, p^*_{h,i}(\lambda)) = -\langle \lambda_n, \mathbf{v}_{h,i} \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega} - \sum_{l=1}^{d-1} \langle \lambda^l_\tau, \mathbf{v}_{h,i} \cdot \boldsymbol{\tau}^l_i \rangle_{\partial\Omega_i \cap \Gamma_{SS}},$$

$$\forall \mathbf{v}_i \in V_{h,i}/\ker\, a_i, \quad (2.12)$$

$$b_i(\mathbf{u}^*_{h,i}(\lambda), w_{h,i}) = 0, \quad \forall\, w_{h,i} \in W_{h,i}, \quad (2.13)$$

where $\{\boldsymbol{\tau}^l_i\}_{l=1}^{d-1}$ is an orthogonal set of unit vectors tangential to $\partial\Omega_i$. Some discussion on handling singular Stokes subdomain problems is given in the next subsection. The complementary set of local problems is to find $(\overline{\mathbf{u}}_{h,i}, \overline{p}_{h,i}) \in V_{h,i} \times W_{h,i}$ such that

$$a_i(\overline{\mathbf{u}}_{h,i}, \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, \overline{p}_{h,i}) = (\mathbf{f}_i, \mathbf{v}_{h,i})_{\Omega_i}, \quad \forall \mathbf{v}_{h,i} \in V_{h,i}/\ker\, a_i, \quad (2.14)$$

$$b_i(\overline{\mathbf{u}}_{h,i}, w_{h,i}) = 0, \quad \forall w_{h,i} \in W_{h,i}. \quad (2.15)$$

Similarly, on each Darcy domain Ω_i, $N_S + 1 \le i \le N$, the first set of local problems is to find $(\mathbf{u}^*_{h,i}(\lambda), p^*_{h,i}(\lambda)) \in V_{h,i} \times W_{h,i}$ with specified interface pressure λ such that

$$a_i(\mathbf{u}^*_{h,i}(\lambda), \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, p^*_{h,i}(\lambda)) = -\langle \lambda, \mathbf{v}_{h,i} \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega}, \quad \forall \mathbf{v}_{h,i} \in V_{h,i}, \quad (2.16)$$

$$b_i(\mathbf{u}^*_{h,i}(\lambda), w_{h,i}) = 0, \quad \forall\, w_{h,i} \in W_{h,i}. \quad (2.17)$$

The corresponding complementary problem is to find $(\overline{\mathbf{u}}_{h,i}, \overline{p}_{h,i}) \in V_{h,i} \times W_{h,i}$ such that

$$a_i(\overline{\mathbf{u}}_{h,i}, \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, \overline{p}_{h,i}) = (\mathbf{f}_i, \mathbf{v}_{h,i})_{\Omega_i}, \quad \forall \mathbf{v}_{h,i} \in V_{h,i}, \quad (2.18)$$

$$b_i(\overline{\mathbf{u}}_{h,i}, w_{h,i}) = -(q_D, w_{h,i})_{\Omega_i}, \quad \forall w_{h,i} \in W_{h,i}. \quad (2.19)$$

Note that for the local Stokes problems (2.12) – (2.13), the boundary conditions on the interfaces Γ_{SS} are of Neumann type:

$$-(\mathbf{T}\mathbf{n}_i) \cdot \mathbf{n}_i = \lambda_n, \quad -(\mathbf{T}\mathbf{n}_i) \cdot \boldsymbol{\tau}^l_i = \lambda^l_\tau, \quad 1 \le l \le d-1, \quad 1 \le i \le N_S,$$

and on the interfaces Γ_{SD} are of Robin type:

$$-(\mathbf{T}\mathbf{n}_i) \cdot \mathbf{n}_i = \lambda_n, \quad -(\mathbf{T}\mathbf{n}_i) \cdot \boldsymbol{\tau}^l_i - \frac{\mu_S \alpha}{\sqrt{K_l}} \mathbf{u}_i \cdot \boldsymbol{\tau}^l_i = 0, \quad 1 \le l \le d-1, \quad 1 \le i \le N_S.$$

For the local Darcy problems (2.16) – (2.17), the boundary conditions on the interfaces $\Gamma_{DD} \cup \Gamma_{SD}$ are of Dirichlet type:

$$p_i = \lambda, \quad N_S + 1 \leq i \leq N.$$

It is easy to verify that problem (2.7) – (2.9) is equivalent to the interface problem for $\lambda_H \in \Lambda_H$:

$$s_H(\lambda_H, \mu_H) := -b_I(\mathbf{u}_h^*(\lambda_H), \mu_H) = b_I(\overline{\mathbf{u}}_h, \mu_H), \quad \forall \mu_H \in \Lambda_H, \tag{2.20}$$

where the above equation follows from the interface condition (2.9) and the global solution can be recovered by

$$\mathbf{u}_h = \mathbf{u}_h^*(\lambda_H) + \overline{\mathbf{u}}_h, \quad p_h = p_h^*(\lambda_H) + \overline{p}_h.$$

Later it will be convenient to write

$$b_I(\mathbf{v}, \mu) = \sum_{i=1}^{N} b_I^i(\mathbf{v}_i, \mu),$$

where

$$b_I^i(\mathbf{v}_i, \mu) = \begin{cases} \langle \mu_n, \mathbf{v}_i \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega} + \sum_{l=1}^{d-1} \langle \mu_\tau^l, \mathbf{v}_i \cdot \boldsymbol{\tau}_i^l \rangle_{\partial\Omega_i \cap \Gamma_{SS}}, & 1 \leq i \leq N_S, \\ \langle \mu_n, \mathbf{v}_i \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega}, & N_s + 1 \leq i \leq N. \end{cases} \tag{2.21}$$

Analysis of the condition number of the reduced problem (2.20) is performed theoretically and numerically in [58].

Theorem 2.3.1. *Assuming $H = O(h)$, there exist positive constants C_1, C_2 independent of h and H, such that for all $\lambda \in \Lambda_H$,*

$$C_1 \frac{K_{min}^2}{K_{max}} (h\|\lambda\|_{\Gamma_{SS}}^2 + \|\lambda\|_{\Gamma_{DD} \cup \Gamma_{SD}}^2) \leq s_H(\lambda, \lambda) \leq C_2 (\|\lambda\|_{\Gamma_{SS}}^2 + \frac{K_{max}}{h} \|\lambda\|_{\Gamma_{DD} \cup \Gamma_{SD}}^2), \tag{2.22}$$

where K_{min} and K_{max} are the minimal and maximal eigenvalues of the permeability $\mathbf{K}$, respectively.

In the case of $H = O(h)$, we also quote the corollary from [58] to show a more concise form of the condition number estimate. Note that $A_i = \text{diam}(\Omega_i)$, $A = \max_{1 \leq i \leq N} A_i$.

Corollary 2.3.2. *The condition number for the algebraic system associated with the coupled Stokes-Darcy flow problem satisfies*

$$\text{cond}(S_h) = O(h^{-1}A^{-1}), \quad \textit{if } \Gamma_{SS} = \varnothing,$$
$$\text{cond}(S_h) = O(h^{-2}A^{-1}), \quad \textit{if } \Gamma_{SS} \neq \varnothing.$$

2.4 ALGEBRAIC INTERPRETATION

Another way to interpret the derivation of the interface problem (2.20) is from the algebraic form of the discretized problem (2.7) – (2.9),

$$\begin{pmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ q \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \begin{pmatrix} M & L^T \\ L & 0 \end{pmatrix} \begin{pmatrix} \xi \\ \lambda \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}, \tag{2.23}$$

where $\xi = (u, p)^T$ is the vector of subdomain unknowns, $r = (f, q)^T$ is the vector of discrete right hand side functions in the coupled system,

$$M = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}, \text{ and } \quad L = \begin{pmatrix} C & 0 \end{pmatrix}.$$

Then by forming the Schur complement of (2.23), we obtain the matrix form of the interface problem (2.20),

$$LM^{-1}L^T\lambda = LM^{-1}r. \tag{2.24}$$

Theorem 2.3.1 implies that the matrix on the left in (2.24) is symmetric and positive definite, and therefore the problem can be solved using a Krylov iterative method such as the conjugate gradient (CG) method. At each iteration we need to evaluate the action of M^{-1}. Since M^{-1} is block-diagonal,

$$M^{-1} = \begin{pmatrix} M_1^{-1} & & \\ & \ddots & \\ & & M_N^{-1} \end{pmatrix},$$

this action requires solving N local subdomain problems, which is done in parallel.

2.5 FLOATING STOKES SUBDOMAINS AND FETI METHOD

As mentioned in Lemma 2.2.2, one issue that arises in these solves is the occurrence of floating subdomains, that is, if a Stokes subdomain Ω_i is surrounded entirely by other Stokes domains, local full Neumann problems occur and the local matrix M_i is singular. In other words, the kernel space ker a_i in (2.12) and (2.14) becomes non-trivial. To handle this issue, we follow the algorithm introduced in [58] which is based on the one-level FETI method proposed by Farhat and Roux [24]. The one-level FETI method can also be viewed as a preconditioned conjugate gradient (PCG) algorithm incorporating an auxiliary coarse problem; see [57] for implementation details. Let $\overline{r}_i$ be orthogonal projection of r onto ker a_i. The algebraic system (2.23) can be written as

$$M\xi + L^T\lambda = r,$$

subject to the constraint

$$L\xi = 0.$$

Let R be a matrix whose columns form or contain a basis for ker(M) and define an operator

$$G = LR.$$

We can split $\lambda = \lambda_0 + \lambda_1$, where

$$\lambda_0 = G(G^TG)^{-1}R^T\overline{r}_S, \quad \text{and} \quad \lambda_1 \in ker(G^T).$$

Due to the block-diagonal structure of G^TG, computing $(G^TG)^{-1}$ requires solving a coarse problem, which can be reduced to solving a local problem of size $k_i \times k_i$ in each subdomain Ω_i, with $k_i = \dim(\ker(M_i))$. Let the operator P be the orthogonal projector onto ker(G^T):

$$P = I - G(G^TG)^{-1}G^T.$$

Applying P^T on both sides of equation (2.24) and letting $\lambda_1 = P\nu$, we derive the projected interface problem

$$P^TLM^{-1}L^TP\nu = P^TL(M^{-1}(\overline{r} - L^T\lambda_0) + M^{-1}(r - \overline{r})). \tag{2.25}$$

Note that in (2.25) there are three actions of M^{-1}, i.e., three local subdomain solves. The two solves on the right hand side both satisfy $\overline{r} - L^T\lambda_0 \perp \ker(M)$ and $r - \overline{r} \perp \ker(M)$, so in the Stokes region they are compatible Neumann solves. On the other hand, for any $P\nu \in \ker(G^T)$, one can verify that $L^T P\nu \perp \ker(M)$, so the local solve on the left hand side is also compatible. Moreover, the matrix $P^T L M^{-1} L^T P$ is symmetric and positive definite in the space $\ker(G^T)$ and therefore (2.25) can be solved with the CG method.

Since M_i may be singular, one must use a pseudoinverse M_i^+. The traditional FETI method uses the Moore-Penrose pseudoinverse, which could be computationally expensive. Since our method involves solving only compatible problems, we can avoid this and in practice we use $M_i^+ := (M_i + \sqrt{\varepsilon} D_i)^{-1}$ to replace the evaluation of M_i^{-1}, where ε is the machine precision and D_i is the velocity mass matrix.

2.6 NUMERICAL EXAMPLES ON MATCHING GRIDS

We present several numerical experiments on matching grids ($H = h$) that illustrate the behavior of the method. In the first test we solve a coupled problem with known analytical solution on different meshes and compute the associated error to verify the convergence of the discretization scheme. In the other tests, which are aimed to examine the convergence of the iterative method, we vary either the mesh size, the permeability, or the number of subdomains. The computational domain is $\Omega = \Omega_S \cup \Omega_D$, where the Stokes domain $\Omega_S = [0,1] \times [\frac{1}{2}, 1]$ and Darcy domain $\Omega_D = [0,1] \times [0, \frac{1}{2}]$. For simplicity we set

$$\mathbf{T}(\mathbf{u}_S, p_S) = -p_S \mathbf{I} + \mu \nabla \mathbf{u}_S$$

in the Stokes equation in Ω_S, and

$$\mathbf{K} = K\mathbf{I}$$

in the Darcy equation in Ω_D, where K is a positive constant. To discretize the system of equations we use the Taylor-Hood [56] triangular finite elements in Ω_S and the lowest order Raviart-Thomas [52] rectangular finite elements in Ω_D. The grid for the discretization in Ω_S

is obtained by first partitioning the domain into rectangles and then dividing each rectangle along its diagonal into two triangles. The grids in Ω_S and Ω_D match on the interface Γ_{SD}.

In our implementation we utilize direct subdomain solvers. This is reasonable, since in practice sufficient number of processors can assure that the subdomain problems are of small to moderate size. As a result the convergence of the interface CG is not affected by inexact subdomain solves. Furthermore, the LU factorization is reused multiple times with different right hand sides at each CG iteration.

For the first test we consider the following analytical solution satisfying the flow equations in Ω_S and Ω_D along with the conditions on the interface Γ_{SD}:

$$\begin{aligned}
\mathbf{u}_S &= \begin{bmatrix} (2-x)(1.5-y)(y-\xi) \\ -\frac{y^3}{3} + \frac{y^2}{2}(\xi+1.5) - 1.5\xi y - 0.5 + \sin(\omega x) \end{bmatrix}, \\
\mathbf{u}_D &= \begin{bmatrix} \omega\cos(\omega x)y \\ \chi(y+0.5) + \sin(\omega x) \end{bmatrix}, \\
p_S &= -\frac{\sin(\omega x)+\chi}{2K} + \mu(0.5-\xi) + \cos(\pi y), \\
p_D &= -\frac{\chi}{K}\frac{(y+0.5)^2}{2} - \frac{\sin(\omega x)y}{K},
\end{aligned}$$

where

$$\mu = 0.1,\ K = 1,\ \alpha_0 = 0.5,\ G = \frac{\sqrt{\mu K}}{\alpha_0}, \xi = \frac{1-G}{2(1+G)},\ \chi = \frac{-30\xi - 17}{48},\ \text{and } \omega = 6.0.$$

The right hand sides $\mathbf{f}_S$, $\mathbf{f}_D$, and q_D for the Stokes-Darcy flow system are obtained by plugging the analytical solution into (1.1), (1.4), and (1.5), respectively. The boundary conditions are as follows: for the Stokes region, the velocity $\mathbf{u}_S$ is specified on the left boundary, and the normal and tangential stresses $(\mathbf{T}\mathbf{n}_S)\cdot\mathbf{n}_S$ and $(\mathbf{T}\mathbf{n}_S)\cdot\boldsymbol{\tau}_S$ are specified on the top and right boundaries; for the Darcy region, the normal velocity $\mathbf{u}_D\cdot\mathbf{n}_D$ is specified on the left boundary and the pressure p_D is specified on the bottom and right boundaries. In this example we use $4\times 4 = 16$ subdomains, 8 in each region. Under these boundary conditions, the six Stokes subdomains on the right are floating: each of the three on the top row has $(u,v) = (1,0)$ and $(0,1)$ as a basis of $\ker(M_i)$, while that basis for each of the three on the second row consists only of $(0,1)$. The computed solution on a mesh with $h = 1/64$ is shown in Figure 2.2. The

Table 2.1: Numerical errors and convergence rates in Ω_S.

h	$\|\mathbf{u}_S - \mathbf{u}_{S,h}\|_{1,\Omega_S}$	$rate$	$\|p_S - p_{S,h}\|_{0,\Omega_S}$	$rate$
1/8	1.69e-01		8.83e-03	
1/16	4.23e-02	2.00	2.23e-03	1.98
1/32	1.05e-02	2.00	5.59e-04	2.00
1/64	2.64e-03	2.00	1.41e-04	1.99
1/128	6.68e-04	1.98	3.59e-05	1.97

contour plots in Figure 2.3 represent the two components of the computed velocity field. We see that the domain decomposition scheme correctly imposes continuity of the normal velocity on Γ_{SD}, but allows for discontinuous tangential velocity across the interface. For our choice of finite element spaces, the error bound in Theorem 2.2.1 predicts second order convergence in the Stokes region and first order convergence in the Darcy region. The results reported in Table 2.1 and Table 2.2 confirm the expected rates. Note that we report the cell-centered L^2-errors $|||\cdot|||$ in the Darcy region, which are superconvergent for both the pressure and the velocity. This is consistent with the theory for mixed finite element methods for Darcy on rectangular grids.

In the next test, for different permeabilities we vary either the mesh size or the number of

Table 2.2: Numerical errors and convergence rates in Ω_D.

h	$\|\|\|\mathbf{u}_D - \mathbf{u}_{D,h}\|\|\|_{\Omega_D}$	$rate$	$\|\|\|p_D - p_{D,h}\|\|\|_{\Omega_D}$	$rate$
1/8	6.04e-02		5.14e-03	
1/16	1.54e-02	1.97	1.29e-03	1.99
1/32	3.88e-03	1.99	3.22e-04	2.00
1/64	9.71e-04	2.00	8.04e-05	2.00
1/128	2.43e-04	2.00	2.02e-05	1.99

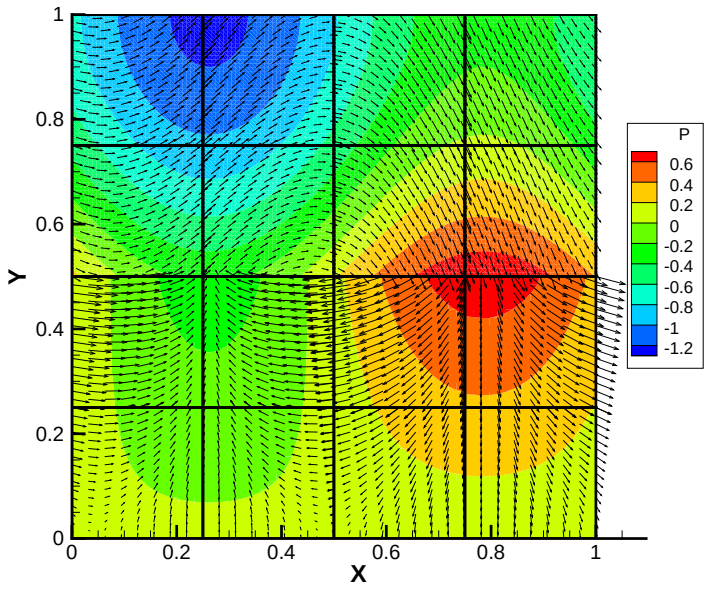

Figure 2.2: Computed solution in the first test on a mesh with $h = 1/64$.

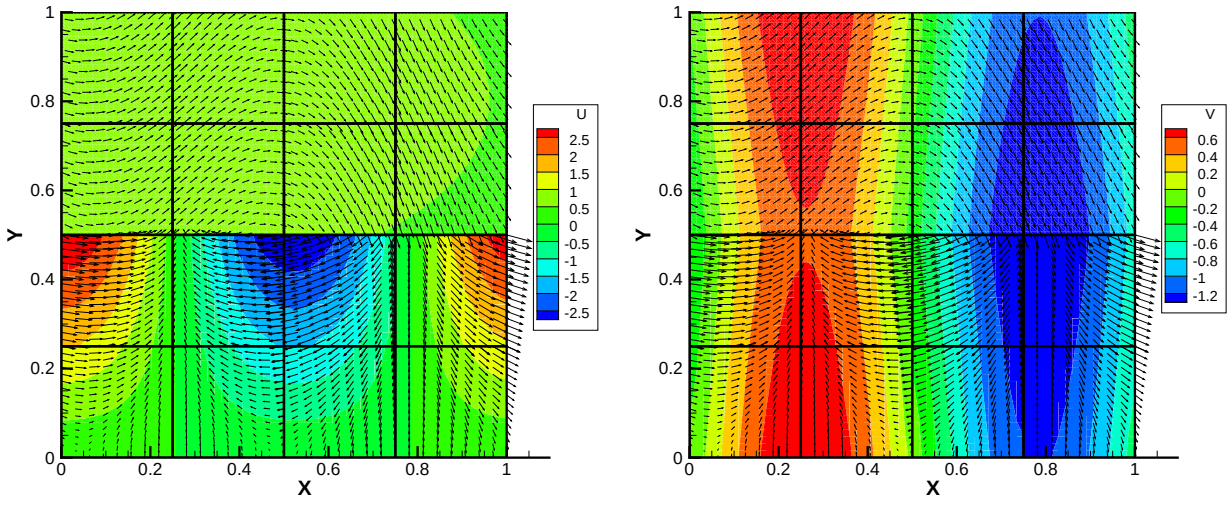

Figure 2.3: Computed velocity field in the first test: horizontal velocity (left); vertical velocity (right).

Table 2.3: Convergence of interface CG: K=1.0, 2 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	4.552	14.637	3.2	7
1/16	4.649	23.064	5.0	11
1/32	4.676	45.317	9.7	17
1/64	4.683	90.479	19.3	24
1/128	4.685	180.958	38.6	35

subdomains to examine the convergence of the iterative method. In Table 2.3 and Table 2.4 we see that on two subdomains, when $h < K$, the minimal eigenvalue of the interface operator does not change much as we refine the mesh, while the maximal eigenvalue changes as $O(h^{-1})$, according to Theorem 2.3.1, which results in condition number of order $O(h^{-1})$. In this case we also see that changing the permeability for a fixed h has no effect on the condition number, which can be explained by the fact that the permeability constants K_{min} and K_{max} appearing in the estimates of the Rayleigh quotient cancel one another when we divide the upper bound by the lower bound. Table 2.5 shows the behavior of the method when $K < h$, in which case both the minimal and the maximal eigenvalues of the interface operator are dominated by constants independent of h, and consequently the condition number does not change significantly as the mesh is refined. This is consistent with Theorem 2.3.1.

In the presence of Stokes-Stokes interfaces, if $h < K$, the bounds in Theorem 2.3.1 imply that the maximal eigenvalue of the interface operator is $O(h^{-1})$ while the minimal is $O(h)$, which means that the condition number is $O(h^{-2})$. This estimate is supported by the results reported in Table 2.6 and Table 2.7. We also see that the largest eigenvalue is doubled when K is doubled, which is consistent with the upper bound in Theorem 2.3.1. For $K < h$, Table 2.8 shows that the maximal eigenvalue does not change when the mesh is refined, confirming the upper bound in Theorem 2.3.1.

Finally we test the effect of the subdomain size on the condition number, running the above tests with $K = 1.0$ on $4 \times 4 = 16$ and $8 \times 8 = 64$ subdomains. Comparing Table 2.6

Table 2.4: Convergence of interface CG: K=2.0, 2 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	8.073	23.930	3.0	8
1/16	8.216	45.477	5.5	12
1/32	8.254	90.490	11.0	17
1/64	8.263	180.918	21.9	25
1/128	8.266	361.902	43.8	36

Table 2.5: Convergence of interface CG: K=0.01, 2 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.302	8.688	28.8	8
1/16	0.263	8.719	33.1	9
1/32	0.270	8.734	32.3	8
1/64	0.278	8.746	31.5	10
1/128	0.280	8.758	31.3	13

Table 2.6: Convergence of interface CG: K=1.0, 4 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.404	25.467	63.1	21
1/16	0.236	50.519	213.7	38
1/32	0.134	100.979	754.8	65
1/64	0.092	202.092	2191.0	95
1/128	0.065	404.432	6183.2	146

Table 2.7: Convergence of interface CG: K=2.0, 4 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.404	50.413	124.8	22
1/16	0.249	100.762	404.8	39
1/32	0.149	201.802	1350.0	66
1/64	0.083	404.096	4874.3	117
1/128	0.084	808.813	9582.9	152

Table 2.8: Convergence of interface CG: K=0.01, 4 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.118	10.846	91.6	19
1/16	0.127	10.856	85.4	19
1/32	0.087	10.858	124.4	22
1/64	0.056	10.859	195.3	27
1/128	0.035	10.860	308.7	39

Table 2.9: Convergence of interface CG: K=1.0, 16 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.287	32.000	111.5	42
1/16	0.125	64.000	512.3	78
1/32	0.062	128.000	2060.6	131
1/64	0.032	256.000	7907.6	236
1/128	0.019	512.000	27115.1	410

on 4 subdomains and Table 2.9 on 16 subdomains, we see that the minimal eigenvalue is approximately proportional to the subdomain size, as expected by the lower bound in Theorem 2.3.1. However, when comparing Table 2.9 and Table 2.10, we notice that the minimal eigenvalue does not change. The reason is that the minimal eigenvalue is controlled by the Stokes region, where it is independent of the subdomain size due the coarse solve we have implemented to handle the local Stokes Neumann problems. This can be observed in Tables 2.11–2.16, where the results for Stokes only and Darcy only problems are reported. In particular, the minimal eigenvalue is smaller for Stokes compared to Darcy on the same mesh and number of subdomains, while the maximal eigenvalue is larger for Darcy. Furthermore, for Stokes the minimal eigenvalue is $O(h)$ and independent of subdomain size, while the maximal eigenvalue is $O(1)$. For Darcy, the minimal eigenvalue is $O(A)$ and the maximal eigenvalue is $O(h^{-1})$. These results are consistent with the theory in Corollary 2.3.2.

Table 2.10: Convergence of interface CG: K=1.0, 64 subdomains.

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.375	31.976	85.3	53
1/16	0.131	64.000	488.2	92
1/32	0.063	128.000	2038.6	152
1/64	0.032	256.000	8035.1	260
1/128	0.019	512.000	27073.1	454

Table 2.11: Convergence of interface CG: K=1.0, 4 subdomains (Stokes only).

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.245	8.037	32.8	20
1/16	0.121	8.087	66.6	29
1/32	0.061	8.099	133.4	36
1/64	0.030	8.102	266.6	47
1/128	0.015	8.103	532.1	58

Table 2.12: Convergence of interface CG: K=1.0, 16 subdomains (Stokes only).

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.251	5.667	22.6	24
1/16	0.123	5.819	47.4	31
1/32	0.061	5.857	96.5	42
1/64	0.030	5.867	193.0	54
1/128	0.015	5.869	385.5	68

Table 2.13: Convergence of interface CG: K=1.0, 64 subdomains (Stokes only).

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	0.222	2.719	12.3	22
1/16	0.124	3.043	24.5	28
1/32	0.061	3.127	51.0	37
1/64	0.030	3.148	103.7	49
1/128	0.015	3.153	207.4	63

Table 2.14: Convergence of interface CG: K=1.0, 4 subdomains (Darcy only).

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	4.051	22.623	5.6	10
1/16	4.061	45.193	11.1	15
1/32	4.063	90.433	22.3	22
1/64	4.064	180.929	44.5	32
1/128	4.064	512.000	126.0	46

Table 2.15: Convergence of interface CG: K=1.0, 16 subdomains (Darcy only).

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	2.166	23.601	10.9	16
1/16	2.172	45.239	20.8	22
1/32	2.174	90.477	41.6	31
1/64	2.174	256.000	117.7	45
1/128	2.174	512.000	235.5	63

Table 2.16: Convergence of interface CG: K=1.0, 64 subdomains (Darcy only).

h	$eig.min.$	$eig.max.$	$cond(S_h)$	$iter.num.$
1/8	1.065	31.361	29.4	26
1/16	1.069	47.773	44.7	31
1/32	1.070	128.000	119.6	44
1/64	1.070	256.000	239.2	61
1/128	1.070	512.000	478.4	87

3.0 MULTISCALE FLUX BASIS FOR STOKES-DARCY FLOWS

In this chapter, we will introduce the details about the implementation of the MMMFEM with multiscale flux basis for Stokes-Darcy flows [30]. As mentioned before, this approach is an extension of the multiscale flux basis implementation of the MMMFEM for Darcy flow developed in [31]. The development of the multiscale flux basis implementation of the MMMFEM for Stokes-Darcy flows involves a number of major technical difficulties compared to the Darcy problem. One issue is the need to solve different types of local problems to compute the basis, Dirichlet in Darcy and Neumann in Stokes. Another difficulty is handling singular full Neumann (i.e. floating) Stokes subdomains. In particular, since the Neumann boundary condition provided by the mortar basis leads to a right hand side that is not orthogonal to the kernel of the subdomain matrix, computing the multiscale flux basis in Stokes involves solving incompatible Neumann problems. However, due to the application of the FETI coarse solve, the multiscale basis is used only to compute the action of the interface operator on compatible data, which is a well-posed Neumann solve. In this chapter we will also show another new development of combining the use of the multiscale flux basis with the balancing preconditioner in the Darcy region [16, 44, 50]. The use of the preconditioner is motivated by the fact that the number of interface iterations is not insignificant for the CPU time, with the cost for computing orthogonal projections and linear combinations, and the inter-processor communication time all playing a role. The balancing preconditioner involves solving Neumann subdomain problems and a coarse problem to exchange global information. It is very efficient and exhibits condition number that grows very mildly with respect to mesh and subdomain size. As a result, the number of interface iterations is reduced significantly, but at the cost of one additional Dirichlet and one Neumann solve per Darcy subdomain per iteration. While the multiscale flux basis can be utilized for the efficient computation of the

extra Dirichlet solves, one needs to compute a new multiscale basis for the Neumann solves. This results in a preconditioned algorithm with the number of local solves independent of the number of interface iterations.

In the numerical examples we compare the computational cost of the implementations with and without multiscale flux basis. Our tests for a wide range of problems show that the multiscale flux basis can improve the efficiency for both unpreconditioned and preconditioned problems.

3.1 ITERATIVE SOLUTION OF THE INTERFACE PROBLEM

In this section, we will present the algorithm of implementing the traditional MMMFEM on the interface problem (2.20) in Section 2.3. For simplicity, we introduce a Steklov-Poincaré type operator $S_H : \Lambda_H \to \Lambda_H$ such that

$$\langle S_H \lambda_H, \mu_H \rangle_\Gamma = s_H(\lambda_H, \mu_H), \quad \forall\, \lambda_H, \mu_H \in \Lambda_H.$$

We note that $S_H \lambda_H = \sum_{i=1}^N S_{H,i} \lambda_{H,i}$, where $S_{H,i} : \Lambda_{H,i} \to \Lambda_{H,i}$ is defined by

$$\langle S_{H,i} \lambda_{H,i}, \mu_{H,i} \rangle_{\Gamma_i} = -b_I^i(\mathbf{u}_{h,i}^*(\lambda_{H,i}), \mu_{H,i}), \quad \forall\, \lambda_{H,i}, \mu_{H,i} \in \Lambda_{H,i}.$$

Let $\mathcal{L}_{h,i} : \Lambda_{H,i} \to V_{h,i}|_{\Gamma_i}$ on $\Gamma_i \cap \Gamma_{SS}$ or $\mathcal{L}_{h,i} : \Lambda_{H,i} \to V_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i}$ on $\Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD})$ be the L^2-orthogonal projection operator from the mortar space onto the (normal) trace of the velocity space in Ω_i. Correspondingly, let $\mathcal{L}_{h,i}^T : V_{h,i}|_{\Gamma_i} \to \Lambda_{H,i}$ or $\mathcal{L}_{h,i}^T : V_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i} \to \Lambda_{H,i}$ be the L^2-orthogonal projection operator from the (normal) trace of the velocity space onto the mortar space. Using the definition (2.21) of $b_I^i(\cdot, \cdot)$, we have that

$$S_{H,i} \lambda_{H,i} = -\mathcal{L}_{h,i}^T \begin{pmatrix} \mathbf{u}_{h,i}^*(\lambda_{H,i}) \cdot \mathbf{n}_i \\ \mathbf{u}_{h,i}^*(\lambda_{H,i}) \cdot \boldsymbol{\tau}_i^l \end{pmatrix} \quad \text{on } \Gamma_i \cap \Gamma_{SS},$$

$$S_{H,i} \lambda_{H,i} = -\mathcal{L}_{h,i}^T \mathbf{u}_{h,i}^*(\lambda_{H,i}) \cdot \mathbf{n}_i \quad \text{on } \Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD}).$$

Problem (2.20) can be rewritten as

$$S_H \lambda_H = g_H, \tag{3.1}$$

where $g_H \in \Lambda_H$ is defined by $\langle g_H, \mu_H \rangle_\Gamma = b_I(\overline{\mathbf{u}}_h, \mu_H), \quad \forall\, \mu_H \in \Lambda_H$. In the general case where floating Stokes subdomains are allowed, we need to solve the corresponding projected interface problem:

$$P^T S_H P \lambda_H = \tilde{g}_H, \tag{3.2}$$

where $\tilde{g}_H$ denotes the right hand side of equation (2.25).

The CG method is applied to solve (3.2), where on each iteration an operator action $P^T S_H P$ on data $\lambda_H \in \Lambda_H$ is computed as follows:

Approach 1: original CG implementation.

1. Project $\lambda_{H,i}$ onto $\ker(G^T)$: $\lambda_{H,i} \to P\lambda_{H,i}$.
2. Project $P\lambda_{H,i}$ onto the local subdomain boundary space: $\gamma_{h,i} = \mathcal{L}_{h,i} P\lambda_{H,i}$.
3. In the Stokes region, solve subdomain problems (2.12)$-$(2.13) with Neumann boundary data $\gamma_{h,i}$. In the Darcy region, solve subdomain problems (2.16) $-$ (2.17) with Dirichlet boundary data $\gamma_{h,i}$. Denote the solutions by $(\mathbf{u}^*_{h,i}(\gamma_{h,i}), p^*_{h,i}(\gamma_{h,i}))$.
4. Project the resulting velocity in Stokes or resulting flux in Darcy back to the mortar space, i.e.,

$$S_{H,i} P\lambda_{H,i} = -\mathcal{L}^T_{h,i} \begin{pmatrix} \mathbf{u}^*_{h,i}(\gamma_{h,i}) \cdot \mathbf{n}_i \\ \mathbf{u}^*_{h,i}(\gamma_{h,i}) \cdot \boldsymbol{\tau}^l_i \end{pmatrix} \quad \text{on } \Gamma_i \cap \Gamma_{SS},$$

   or

$$S_{H,i} P\lambda_{H,i} = -\mathcal{L}^T_{h,i} \mathbf{u}^*_{h,i}(\gamma_{h,i}) \cdot \mathbf{n}_i \quad \text{on } \Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD}),$$

   and compute the jump across all subdomain interfaces:

$$S_H P\lambda_H = \sum_{i=1}^{N} S_{H,i} P\lambda_{H,i}.$$

5. Apply P $(= P^T)$ to project the jump onto $\ker(G^T)$: $S_H P\lambda_H \to P^T S_H P\lambda_H$.

3.2 MULTISCALE FLUX BASIS IMPLEMENTATION

Notice that the dominant computational costs in the above algorithm comes from the subdomain solves in step 3. Thus, for the original implementation of the MMMFEM, the total

number of solves in each subdomain is approximately equal to the number of CG iterations. Even though all subdomain solves are computed in parallel, this can can still be very costly when the condition number of the interface problem is large due to a highly refined mesh.

In this section we introduce the notion of a multiscale flux basis, following the idea from [31]. Our primary motivation is to improve the efficiency of the solution of the interface problem (3.2). This approach aims to eliminate the dependency between the total number of solves and the number of CG iterations. In order to achieve this, we precompute and store the flux or velocity subdomain responses, called multiscale flux basis, associated with each coarse scale mortar degree of freedom on every Darcy or Stokes subdomain. This requires solving a fixed number of subdomain solves. Then, the solution of subdomain problems on each CG iteration is replaced by linear combinations of the multiscale flux basis functions. As a result, the total number of solves per subdomain is independent of the number of CG iterations and thus insensitive to refining the subdomain grids.

In subdomain Ω_i, $i = 1, 2, \ldots N$, let $\mathcal{N}_i$ be the number of degrees of freedom in the mortar space $\Lambda_{H,i}$ on Γ_i, and let $\left\{\xi_{H,i}^{\{k\}}\right\}_{k=1}^{\mathcal{N}_i}$ be a basis of $\Lambda_{H,i}$. Any $\lambda_{H,i} \in \Lambda_{H,i}$ can be expressed as $\lambda_{H,i} = \sum_{k=1}^{\mathcal{N}_i} \alpha_i^{\{k\}} \xi_{H,i}^{\{k\}}$. Define the multiscale basis $\left\{\phi_{H,i}^{\{k\}}\right\}_{k=1}^{\mathcal{N}_i} \subset \Lambda_{H,i}$ as

$$\phi_{H,i}^{\{k\}} = S_{H,i} \xi_{H,i}^{\{k\}}, \quad k = 1, \ldots, \mathcal{N}_i.$$

The action of the interface operator on any mortar function can then be computed as

$$S_{H,i}\lambda_{H,i} = S_{H,i}\left(\sum_{k=1}^{\mathcal{N}_i} \alpha_i^{\{k\}} \xi_{H,i}^{\{k\}}\right) = \sum_{k=1}^{\mathcal{N}_i} \alpha_i^{\{k\}} S_{H,i} \xi_{H,i}^{\{k\}} = \sum_{k=1}^{\mathcal{N}_i} \alpha_i^{\{k\}} \phi_{H,i}^{\{k\}},$$

which is simply a linear combination of the multiscale basis.

The algorithm for computing the multiscale basis $\left\{\phi_{H,i}^{\{k\}}\right\}_{k=1}^{\mathcal{N}_i} \subset \Lambda_{H,i}$ on a subdomain Ω_i is as follows:

Approach 2: computation of multiscale flux basis.

For $k = 1, \ldots, \mathcal{N}_i$

1. Project the mortar basis function onto the local subdomain boundary space:

$$\eta_{h,i}^{\{k\}} = \mathcal{L}_{h,i} \xi_{H,i}^{\{k\}}.$$

2. If Ω_i is a Stokes subdomain, solve the subdomain problem (2.12) – (2.13) with Neumann boundary data $\eta_{h,i}^{\{k\}}$. If it is Darcy, solve the subdomain problem (2.16) – (2.17) with Dirichlet boundary data $\eta_{h,i}^{\{k\}}$. Denote the solutions by $(\mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}}), p_{h,i}^*(\eta_{h,i}^{\{k\}}))$.
3. Project the resulting velocity in Stokes or resulting flux in Darcy back to mortar space, which gives the multiscale flux basis:

$$\phi_{H,i}^{\{k\}} = -\mathcal{L}_{h,i}^T \begin{pmatrix} \mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}}) \cdot \mathbf{n}_i \\ \mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}}) \cdot \boldsymbol{\tau}_i^l \end{pmatrix} \quad \text{on } \Gamma_i \cap \Gamma_{SS},$$

or

$$\phi_{H,i}^{\{k\}} = -\mathcal{L}_{h,i}^T \mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}}) \cdot \mathbf{n}_i \quad \text{on } \Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD}).$$

The multiscale flux basis can be used in the conjugate gradient method for solving (3.2). In every iteration, the operator action $P^T S_H P$ on any $\lambda_H \in \Lambda_H$ is computed with the following steps:

Approach 3: CG with multiscale basis.

1. Project $\lambda_{H,i}$ onto $\ker(G^T)$: $\lambda_{H,i} \to P\lambda_{H,i}$.
2. Denote by $\left\{c_i^{\{k\}}\right\}_{k=1}^{\mathcal{N}_i}$ the expansion coefficients of $P\lambda_{H,i}$ in the mortar basis:

$$P\lambda_{H,i} = \sum_{k=1}^{\mathcal{N}_i} c_i^{\{k\}} \xi_{H,i}^{\{k\}}.$$

3. Use a linear combination of the multiscale flux basis to compute the resulting velocity (if Ω_i is Stokes) or flux (if Ω_i is Darcy):

$$S_{H,i} P\lambda_{H,i} = S_{H,i} \left(\sum_{k=1}^{\mathcal{N}_i} c_i^{\{k\}} \xi_{H,i}^{\{k\}} \right) = \sum_{k=1}^{\mathcal{N}_i} c_i^{\{k\}} S_{H,i} \xi_{H,i}^{\{k\}} = \sum_{k=1}^{\mathcal{N}_i} c_i^{\{k\}} \phi_{H,i}^{\{k\}}.$$

and compute the jump across all subdomain interfaces:

$$S_H P\lambda_H = \sum_{i=1}^{N} S_{H,i} P\lambda_{H,i}.$$

4. Apply P $(= P^T)$ to project the jump onto $\ker(G^T)$: $S_H P\lambda_H \to P^T S_H P\lambda_H$.

Note that in the computation of the multiscale flux basis in Step 1 of Approach 2, the right hand side for the local Neumann solve in Stokes has not been projected to $\ker(G^T)$.

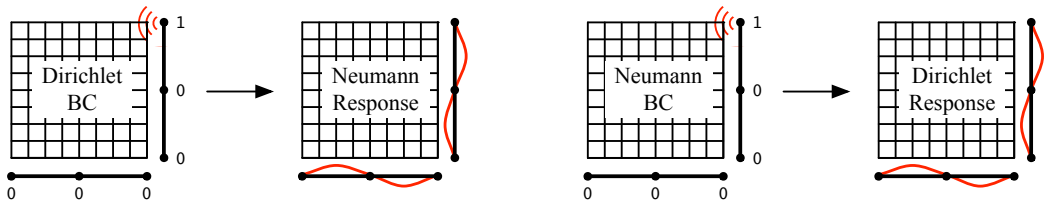

Figure 3.1: Computation of the multiscale flux basis in Darcy (left) and Stokes (right).

However, in Approach 3, the multiscale basis is used only for computing the action of $S_{H,i}$ on $P\lambda_{H,i}$, which is a well-posed local Neumann solve. As a result, the computation of $S_{H,i}P\lambda_{H,i}$ via the linear combination of the multiscale flux basis in Approach 3 gives an equivalent result to computing $S_{H,i}P\lambda_{H,i}$ directly by solving a well posed local Neumann problem with data $P\lambda_{H,i}$. In addition, since P is linear, applying it on the computed jump in the last step is equivalent to projecting the multiscale flux basis to $\ker(G^T)$.

Figure 3.1 (left) illustrates one pressure mortar basis function in a Darcy domain and the computed flux response, which is the corresponding multiscale flux basis function. Similarly, Figure 3.1 (right) shows one normal stress mortar basis function in a Stokes domain and the computed velocity response. Comparing the new algorithm with the original MMMFEM, we notice that there are no subdomain solves in the CG iterations. The dominant cost now shifts to the computation of a multiscale flux basis, which depends on the number of mortar degrees of freedom. Since the mortar space is on the coarse scale, this cost is relatively small and independent of the fine grid. Furthermore, unlike other multiscale methods such as the variational multiscale method or the multiscale finite element method, where the fine scale solution on the entire coarse element needs to be stored, our method requires storing only coarse scale interface data - the flux or velocity response. Therefore the extra storage cost is significantly smaller compared to existing methods.

Even though the multiscale flux basis algorithm does not require subdomain solves in the CG iterations, the number of interface iterations is not insignificant. Some of the cost is due to the time needed to compute the orthogonal projections and linear combinations, but the more significant cost comes from inter-processor communication. It is therefore possible

to reduce the overall computational cost by applying a preconditioner for the solution of the interface problem (3.2) in order to decrease its condition number, which results in decreasing the number of interface iterations. For the performance comparison in the numerical examples in the next section, we employ in the Darcy region the balancing preconditioner introduced in [16, 44, 50]. This preconditioner involves solving Neumann subdomain problems and a coarse problem to exchange global information. It is very efficient and exhibits condition number that grows very mildly with respect to h and H. We do not apply a preconditioner in the Stokes region, although due the coarse solver, the condition number in the Stokes region is insensitive to the subdomain size [58].

Let M_D be the Darcy component of M, which is the block-diagonal matrix with blocks M_i, $i = N_S+1, \ldots, N$, and let Λ_H^D be the restriction of Λ_H to $\Gamma_{DD} \cup \Gamma_{SD}$. In algebraic form, the balancing preconditioner can be expressed as

$$B_{bal}^{-1} = \sum_{i=N_S+1}^{N} L M_i^+ L^T,$$

where M_i^+ is the Moore-Penrose pseudo-inverse of M_i. The detailed algorithm is shown below. Define a partition of unity D_i such that

$$\sum_{i=N_S+1}^{N} D_i \lambda = \lambda, \quad \forall \lambda \in \Lambda_H^D,$$

and define a coarse space

$$Z = \left\{ \lambda \in \Lambda_H^D : \lambda = \sum_{i=N_S+1}^{N} D_i \psi_i, \ \psi_i \in Z_i \right\},$$

where Z_i are spaces of constant vectors such that $\ker(M_i) \subseteq Z_i$, $i = N_S+1, \ldots, N$. In every CG iteration, given residual $r \in \Lambda_H^D$, compute $B_{bal}^{-1} r$ as follows:

Approach 4: balancing preconditioner.

1. Solve a coarse problem: find $\tilde{r} \in Z$ such that

$$a_i(\tilde{r}, \mu) = \langle r, \mu \rangle, \quad \forall \mu \in Z,$$

and balance the residual:

$$r^{bal} = r - M_D \tilde{r}. \tag{3.3}$$

2. Distribute r^{bal} to subdomains:

$$r_i = D_i^T r^{bal}.$$

3. Solve local Neumann problem for $\lambda_i \in \Lambda_{H,i}$:

$$M_i \lambda_i = r_i \tag{3.4}$$

4. Average the local solutions:

$$\lambda = \sum_{i=N_S+1}^{N} D_i \lambda_i.$$

5. Solve a coarse problem for $\tilde{\lambda} \in Z$:

$$a_i(\tilde{\lambda}, \mu) = \langle r^{bal}, \mu \rangle - a_i(\lambda, \mu), \quad \forall \mu \in Z,$$

and update the local solutions:

$$B_{bal}^{-1} r = \lambda + \tilde{\lambda} + \tilde{r}.$$

In the next section we study numerically the efficiency of the multiscale flux basis implementation by comparing four different methods based on the above approaches:

- Method 1: the original MMMFEM with no preconditioner, Approach 1,
- Method 2: the original MMMFEM with balancing preconditioner, Approachs 1 and 4,
- Method 3: multiscale flux basis implementation of the MMMFEM with no preconditioner, Approaches 2 and 3,
- Method 4: multiscale flux basis implementation of the MMMFEM with balancing preconditioner, Approaches 2, 3, and 4.

The maximal number of local solves per subdomain for each method is given in Table 3.1, where N_{iter} is the total number of CG iterations. The number of solves in Method 1 in each subdomain is $N_{iter}+3$, since there is one solve per CG iteration and 3 extra solves for setting up the right hand side of (2.25) in Stokes and the recovery of the solution $(\mathbf{u}, p)$. In Method 2, there are two extra subdomain solves in each CG iteration in Darcy for the balancing preconditioner, one Dirichlet solves in (3.3) and one Neumann solve in (3.4), and at most 10 extra solves for the setup of the balancing preconditioner [50]. In Method 3, the maximal

Table 3.1: Maximal number of solves per subdomain for each method.

Method 1	$N_{iter} + 3$
Method 2	$3N_{iter} + 10$
Method 3	$\max\{\mathcal{N}_i\}_{i=1}^{N} + 3$
Method 4	$\max\{\max\{\mathcal{N}_i\}_{i=1}^{N_S} + 3,\ 2\max\{\mathcal{N}_i\}_{i=N_S+1}^{N} + 8\}$

number of solves is given by $\max\{\mathcal{N}_i\}_{i=1}^{N} + 3$, which includes one solve in the computation of each multiscale flux basis, plus 3 extra solves for setting up the right hand side of (2.25) in Stokes and the recovery of the solution $(\mathbf{u}, p)$. In Method 4, the maximal number of solves in Stokes subdomains is the same as in Method 3, which is $\max\{\mathcal{N}_i\}_{i=1}^{N_S} + 3$. In Darcy subdomains, two different sets of multiscale basis are computed for both the Dirichlet solves in each CG iteration and the Neumann solves (3.4) in the balancing preconditioner, and these solves are replaced by a linear combination of the corresponding multiscale basis. Therefore, the maximal number of solves in the Darcy region is $2\max\{\mathcal{N}_i\}_{i=N_S+1}^{N} + 8$. We note that in the original implementation, Method 1 and Method 2, the number of subdomain solves is proportional to the number of CG iterations, while in the multiscale flux basis Method 3 and Method 4, the number of solves depends only on the number of local mortar degrees of freedom. Furthermore, the balancing preconditioner significantly reduces the number of CG iterations, which results in reduced computational time. In the examples below we compare all four methods and identify the most efficient method in terms of maximal number of solves per subdomain and computational time. We emphasize the comparison between the unpreconditioned Methods 1 and 3, as well as the preconditioned Methods 2 and 4 for better demonstration of the multiscale flux implementation.

Remark 3.2.1. *In the numerical experiments we report both the number of subdomain solves and the CPU times. While the former is accepted as an objective measure of the computational efficiency of domain decomposition methods, we include the latter in order to provide a more complete picture of the total cost. We note that CPU runtime is highly machine*

dependent. In particular, interprocessor communication cost may play a significant role. As a result, the gains we observe in CPU time are smaller relative to the gains in number of solves. Communication cost can be reduced by exploring redundancy and shared multi-core memory architectures, which is beyond the scope of the thesis.

3.3 NUMERICAL EXAMPLES

In this section we present three numerical tests to illustrate the efficiency of the multiscale flux basis implementation by comparing the maximal number of subdomain solves and total runtime for Methods 1-4. In all examples, the lowest order Taylor-Hood triangular finite elements are used in Stokes and the lowest order Raviart-Thomas rectangular finite elements are used in Darcy. Discontinuous piecewise linear mortar finite elements are used for all mortars spaces on subdomain interfaces. We take $\mathbf{T}(\mathbf{u}_S, p_s) = -p_s\mathbf{I} + 2\mu_S\nabla\mathbf{u}_S$, $\mu_S = \mu_D = \mu$ and $\mathbf{K} = K\mathbf{I}$, where K is a uniformly positive scalar function. All four methods produce the same solution, within the relative convergence tolerance 10^{-6}, except in the last test case the tolerance is 10^{-9}. We have previously performed extensive verification studies of the code, including testing the convergence of the numerical solution to the true solution as the grids are refined, see [58]. The results indicate convergence of order predicted by Theorem 2.2.1. One of the cases tested was the smooth solution in Example 1 below. The numerical tests presented in this section are run on a parallel cluster of Intel Xeon CPU E5-2650 v3 @ 2.30GHz processors with 192GB RAM. The problems are solved in parallel such that each subdomain is assigned to one core.

3.3.1 Example 1: Regular shape domain with smooth solution

In this example, the domain is the unit square with $\Omega_S = (0,1) \times (0.5,1)$ and $\Omega_D = (0,1) \times (0,0.5)$. The problem has a given true solution such that

$$\mathbf{u}_S = \begin{bmatrix} (2-x)(1.5-y)(y-\xi) \\ -\frac{y^3}{3} + \frac{y^2}{2}(\xi+1.5) - 1.5\xi y - 0.5 + \sin(\omega x) \end{bmatrix},$$
$$\mathbf{u}_D = \begin{bmatrix} \omega\cos(\omega x)y \\ \chi(y+0.5) + \sin(\omega x) \end{bmatrix},$$
$$p_S = -\frac{\sin(\omega x)+\chi}{2K} + \mu(0.5-\xi) + \cos(\pi y),$$
$$p_D = -\frac{\chi}{K}\frac{(y+0.5)^2}{2} - \frac{\sin(\omega x)y}{K},$$

where

$$\mu = 0.1,\ K = 1,\ \alpha = 0.5,\ G = \frac{\sqrt{\mu K}}{\alpha},\ \xi = \frac{1-G}{2(1+G)},\ \chi = \frac{-30\xi - 17}{48},\ \text{and } \omega = 6.0.$$

The right hand sides $\mathbf{f}_S$, $\mathbf{f}_D$ and q_D in the Stokes-Darcy problem are computed using the given exact solution. The problem is solved by the four methods using four different levels of domain decomposition: 2×2, 4×4, 6×6, 8×8 and 10×10 subdomains. In any level, each subdomain is discretized by a 10×10 or 4×4 local mesh in a "checkerboard" manner, i.e., no neighboring subdomains have the same mesh. The mortar mesh on each subdomain interface is 2×1. The outside boundary conditions are given as follows: for Darcy, the left boundary is of Neumann type and the bottom and right boundaries are of Dirichlet type; for Stokes, the left boundary is of Dirichlet type and the top and right boundaries are Neumann type. Under this setting, all Stokes subdomains except the leftmost column are floating, with $\ker M_i = \text{span}\{(0,1)\}$ if the domain has Stokes-Darcy interface on the bottom and $\ker M_i = \text{span}\{(1,0),(0,1)\}$ otherwise. The test results for Example 1 are shown in Table 3.2 and Table 3.3. In particular, number of CG iterations, maximal number of subdomain solves, and maximal CPU time per subdomain are reported in Table 3.2, indicating the highest workload of all CPUs in each parallel run. We note that the numbers match Table 3.1. The minimal and maximal eigenvalues and condition number are reported in Table 3.3. The computed solution with $4 \times 4 = 16$ subdomains is shown in Figure 3.2.

Table 3.2: Example 1 test results

	Method 1			**Method 2**		
Subdomains	N_{iter}	Solves	Time(s)	N_{iter}	Solves	Time(s)
$2 \times 2 = 4$	38	41	0.587	22	74	0.607
$4 \times 4 = 16$	121	124	1.000	53	169	0.876
$6 \times 6 = 36$	182	185	1.566	78	244	1.341
$8 \times 8 = 64$	230	233	3.261	94	292	2.523
$10 \times 10 = 100$	280	283	7.352	98	304	4.764
	Method 3			**Method 4**		
Subdomains	N_{iter}	Solves	Time(s)	N_{iter}	Solves	Time(s)
$2 \times 2 = 4$	38	**27**	**0.579**	22	38	0.583
$4 \times 4 = 16$	121	**59**	0.808	53	72	**0.695**
$6 \times 6 = 36$	182	**67**	1.245	78	72	**1.151**
$8 \times 8 = 64$	230	**67**	2.328	94	72	**2.102**
$10 \times 10 = 100$	280	**67**	4.170	98	72	**3.392**

Bold denotes the smallest number of subdomain solves and the fastest run

Table 3.3: Example 1 condition numbers

	Method 1, 3			**Method 2, 4**		
Subdomains	eig.min.	eig.max.	cond.num.	eig.min.	eig.max.	cond.num.
$2 \times 2 = 4$	0.349	36.9	105.7	0.409	10.730	26.2
$4 \times 4 = 16$	0.134	90.0	670.8	0.134	30.409	226.3
$6 \times 6 = 36$	0.082	134.7	1635.4	0.082	32.135	393.1
$8 \times 8 = 64$	0.062	179.6	2881.6	0.061	32.108	527.1
$10 \times 10 = 100$	0.050	224.5	4502.6	0.050	31.973	644.3

Since the mesh size h decreases as we increase the number of subdomains, the interface condition number increases. This is evident from Table 3.3. Methods 2 and 4, which employ the balancing preconditioner in the Darcy region, control the maximal eigenvalue and result in a more modest increase in the condition number, compared to the unpreconditioned Methods 1 and 3. This has an effect on the number of CG iterations reported in Table 3.2, with a much sharper increase in N_{iter} for Methods 1 and 3. Regarding the number of subdomain solves, we observe that the multiscale flux basis Methods 3 and 4 have reduced number of solves compared to Methods 1 and 2. Method 3 has the smallest number of

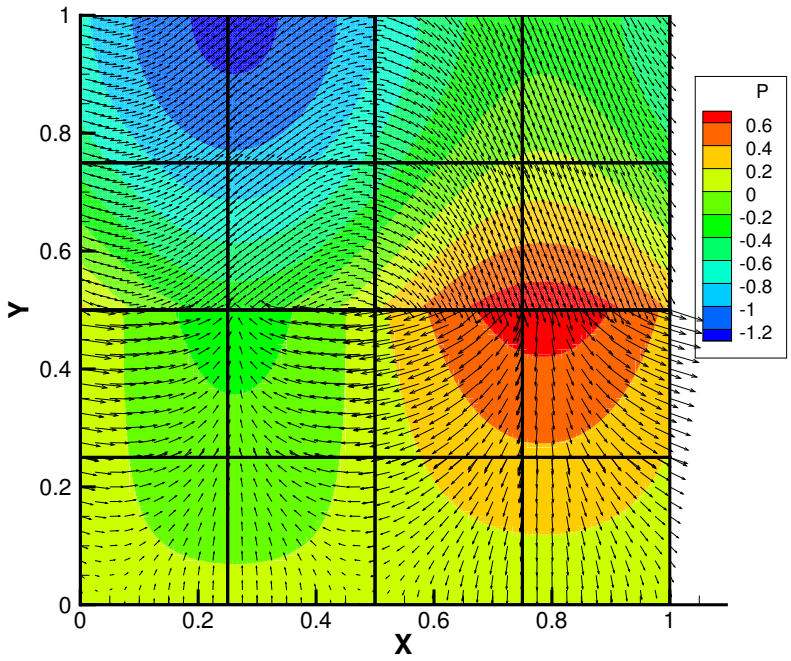

Figure 3.2: Example 1, 16 subdomains, solution

subdomain solves in all cases (marked in bold). Also Method 3 and 4 only involve a fixed number of solves for precomputing the multiscale flux basis, which depend on the local number degrees of freedom of the mortar space. Note that the number of solves for both methods stay fixed from the 36 domains case to the 100 domains case, and will not change if more subdomains are added, since there are no more new mortar degrees of freedom created per subdomain. In Method 4, the balancing preconditioner costs one extra local Neumann solve per degree of freedom in the precomputation step for each Darcy subdomain. However, the significantly reduced number of iterations results in reduced communication cost and savings in the overall computational time. We mark in bold the fastest run for each case in Table 3.2, which is achieved by either Method 3 or 4 in all four levels. It is not easy to make a fair comparison between Method 3 and 4, since the former reduces the number of solves, while the latter reduces the number of iterations at the cost of extra solve per iteration.

Nevertheless, both methods are more efficient than Methods 1 and 2.

3.3.2 Example 2: Irregular shape domain with heterogeneous permeability

The second example is a Stokes-Darcy problem with heterogeneous permeability field on an irregularly shaped domain. The domain is roughly contained in the $[0, 2] \times [0, 1]$ rectangular region, see Figure 3.4, with he Stokes region in the top half and the Darcy region at the bottom. Following [60, 55], we handle the irregular geometry by the multipoint flux mixed finite element method for the pressure in Darcy and standard conforming elements in Stokes. We also impose the mortar conditions on curved interfaces by mapping the physical grids to reference grids with flat interfaces, see [55].

The heterogeneous permeability is given by a single realization of a stochastic permeability field K, which can be generated by a sum of the local Karhunen-Loève (KL) expansion [33]. Let $Y = \ln(K)$ and let Y' be defined by

$$Y'(\mathbf{x}, \omega) := Y(\mathbf{x}, \omega) - E[Y](\mathbf{x}),$$

where $E[Y](\mathbf{x})$ stands for the expectation function. Denote the series expansion of the covariance function of Y as

$$C_Y(\mathbf{x}, \overline{\mathbf{x}}) = \sum_{j=1}^{\infty} \lambda_j f_j(\mathbf{x}) f_j(\overline{\mathbf{x}}).$$

Then the KL-expansion of Y' with N_{term} terms is given as

$$Y'(\mathbf{x}, \omega) \approx \sum_{j=1}^{N_{term}} \xi_j(\omega) \sqrt{\lambda_j} f_j(\mathbf{x}),$$

where ξ_j are normal random variables with zero mean and unit variance. In this example, the covariance function is specified as

$$C_Y(\mathbf{x}, \overline{\mathbf{x}}) = \sigma_Y^2 \exp\left[\frac{-|x_1 - \overline{x}_1|}{\eta_1} - \frac{|x_2 - \overline{x}_2|}{\eta_2}\right]$$

with $\sigma_Y = 2.1$, $\eta_1 = 0.1$, $\eta_2 = 0.05$ and $N_{term} = 400$. For the mean value we set $E[Y](\mathbf{x}) = 1.0$. A plot for the permeability realization is shown in Figure 3.3.

All methods are tested on four different levels of domain decompositions: 4×2, 8×4, 12×6 and 16×8. The local meshes are 18×15, 15×12 "checkerboard" in Darcy; and $12 \times 15, 9 \times 12$ "checkerboard" in Stokes. The mortar mesh is 4×1 on each subdomain interface. The outside boundary conditions are given as follows. In Darcy, no flow condition is specified on the left and right boundaries, with Dirichlet condition at the bottom. In Stokes, we specify inflow condition on the left boundary, zero flow condition on the right, and zero stress condition on the top.

Table 3.4 shows the results using Methods 1 - 4, and Table 3.5 shows the corresponding eigenvalues and condition numbers. Due to the size of K_{max} and a small h as the number of subdomain grows, the problem has a huge condition number and requires thousands of iterations for CG to converge in original MMMFEM implementation. With the multiscale flux basis, the number of solves in Method 3 is significantly smaller (about 90 percent) than Method 1, since it is independent of N_{iter}. However, the saves in runtime is not as significant. This is because the communication cost in distributed parallel computers becomes a big factor when the number of subdomains is large. As a results, even though Method 3 has the smallest number of solves, it is slower than Method 4 in the last three cases. We note that the balancing preconditioner is very effective in this situation. The maximal eigenvalue is controlled effectively and so is the number of CG iterations, as seen in Table 3.5. The gain in runtime due to the balancing preconditioner becomes more significant when increasing the number of subdomain.

3.3.3 Example 3: Adaptive mesh in Darcy

The third test case illustrates how the multiscale flux basis method reduces the computational cost when adaptive mesh refinement is used in the Darcy region. In this case, the permeability field K in the Darcy region is also generated from a single realization of a stochastic field using the KL expansion and it is highly heterogeneous. The KL parameters are correlation lengths $\eta_1 = 0.25$ and $\eta_2 = 0.125$, mean value $E[Y](\mathbf{x}) = 2.0$, variance $\sigma_Y = 2.1$ and $N_{term} = 400$. The generated permeability field is shown in Figure 3.5. The domain is the unit square with the Stokes and Darcy regions in the top and bottom half, respectively. The

Table 3.4: Example 2 test results

	Method 1			**Method 2**		
Subdomains	N_{iter}	Solves	Time(s)	N_{iter}	Solves	Time(s)
$4 \times 2 = 8$	220	223	1.165	66	207	0.979
$8 \times 4 = 32$	1069	1072	6.312	194	592	2.303
$12 \times 6 = 72$	1525	1528	25.791	239	727	7.002
$16 \times 8 = 128$	2375	2378	90.088	301	913	16.261
	Method 3			**Method 4**		
Subdomains	N_{iter}	Solves	Time(s)	N_{iter}	Solves	Time(s)
$4 \times 2 = 8$	220	**83**	**0.850**	67	103	0.901
$8 \times 4 = 32$	1069	**115**	2.260	194	136	**1.733**
$12 \times 6 = 72$	1525	**131**	6.771	239	136	**3.754**
$16 \times 8 = 128$	2375	**131**	24.407	301	136	**9.828**

Bold - denotes the smallest number of subdomain solves and the fastest run

Table 3.5: Example 2 condition numbers

	Method 1, 3			**Method 2, 4**		
Subdomains	eig.min.	eig.max.	cond.num.	eig.min.	eig.max.	cond.num.
$4 \times 2 = 8$	0.114	416.2	3637.2	0.112	26.368	236.3
$8 \times 4 = 32$	0.025	2228.8	88015.3	0.023	44.044	1904.4
$12 \times 6 = 72$	0.017	2371.0	143620.6	0.016	38.682	2391.8
$16 \times 8 = 128$	0.013	4444.5	351727.1	0.012	39.937	3325.1

boundary conditions for Darcy are no flow condition on the left and right with Dirichlet on the bottom. The boundary conditions for Stokes are inflow condition on the left and zero stress on the right, with specified horizontal velocity and zero normal stress on the top. The problem is solved on a 6×6 domain decomposition.

In this test, an adaptive mesh refinement algorithm [4, 61] computes *a posteriori* error indicators in the Darcy region and refines the local and mortar grids for subdomains with large error. The meshes in the Stokes region and on the Stokes-Darcy interface are not affected. We set a starting local mesh of 4×4 in all subdomains and a uniform mortar mesh of 2×1 on every edge. We again run and compare all four methods as in the previous two

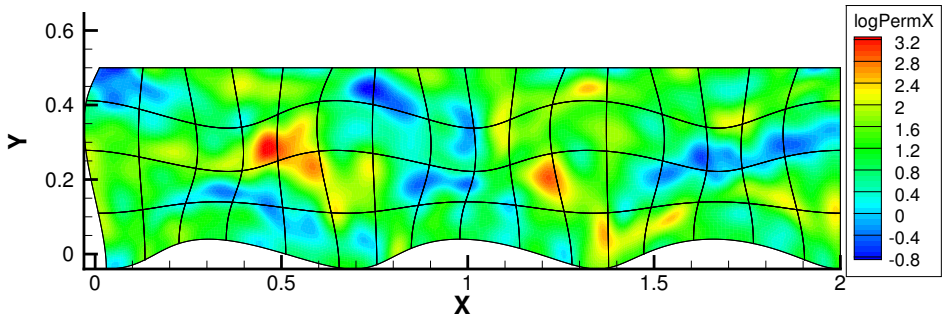

Figure 3.3: Example 2, 128 subdomains, permeability

examples. Table 3.6 reports the sum of numbers of CG iterations in all levels as well as the maximal number of solves and runtime.

The velocity computed by Method 3 on the last mesh refinement level is presented in Figure 3.6. It is easy to observe that regions with higher heterogeneity are refined more times. The total number of solves after the last refinement on each subdomain for Method 1 and 3 are shown in Figure 3.7 and Figure 3.8, respectively. Without multiscale flux basis, the number of solves is the same in all Darcy subdomains, regardless of the difference of their meshes. Stokes domains report the same numbers. with just four extra solves for implementing the FETI method. With multiscale flux basis, the number of solves depends on the mortar degrees of freedom on the surrounding interfaces. Hence, with different levels of mesh refinement, the Darcy subdomains are reporting different number of solves. Clearly the number of solves is reduced significantly in Method 3 compared to Method 1. It is evident from the left column of Table 3.6, that the multiscale flux basis saves roughly 95 percent in

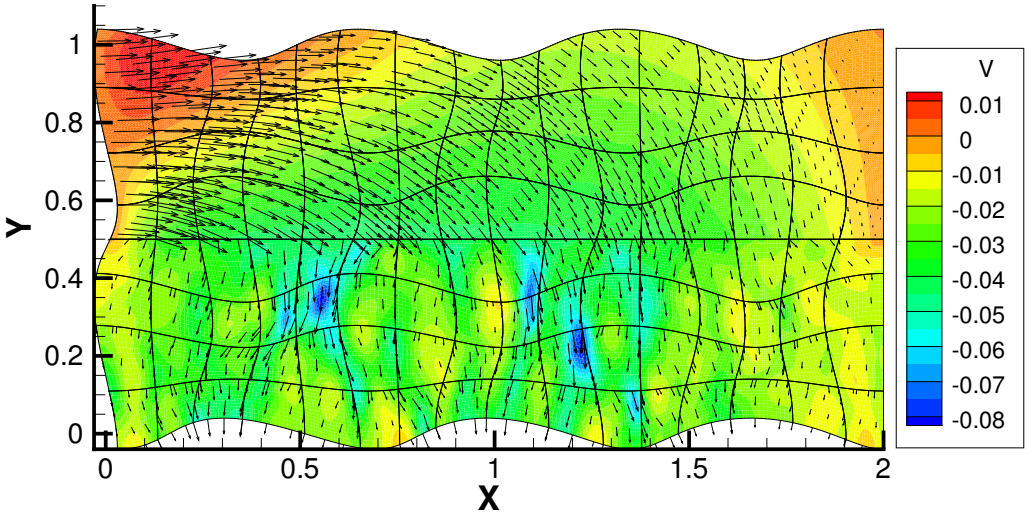

Figure 3.4: Example 2, 128 subdomains, velocity solution

maximum solves and 30 percent in runtime. Comparing results in the right column, Methods 4 saves roughly 60 percent in maximum solves and 20 percent in runtime from Method 2. If there were more levels of mesh refinement, there would be more significant saving in both numbers, since the computational saving due to the multiscale flux basis occurs on every refinement level.

We also comment that if the communication cost is reduced in both implementations with and without multiscale flux basis, the relative savings due to the reduced number of solves will be much larger. To improve the CPU time, one needs to explore shared multi-core memory architectures which is beyond the scope of numerical analysis.

Table 3.6: Example 3 with adaptive mesh refinement, computational results

Method 1			**Method 2**		
N_{iter}	Max Solves	Time(s)	N_{iter}	Max Solves	Time(s)
9715	9733	11.10	823	2481	5.57
Method 3			**Method 4**		
N_{iter}	Max Solves	Time(s)	N_{iter}	Max Solves	Time(s)
9715	**488**	7.85	823	988	**4.66**

Bold - denotes the smallest number of subdomain solves and the fastest run

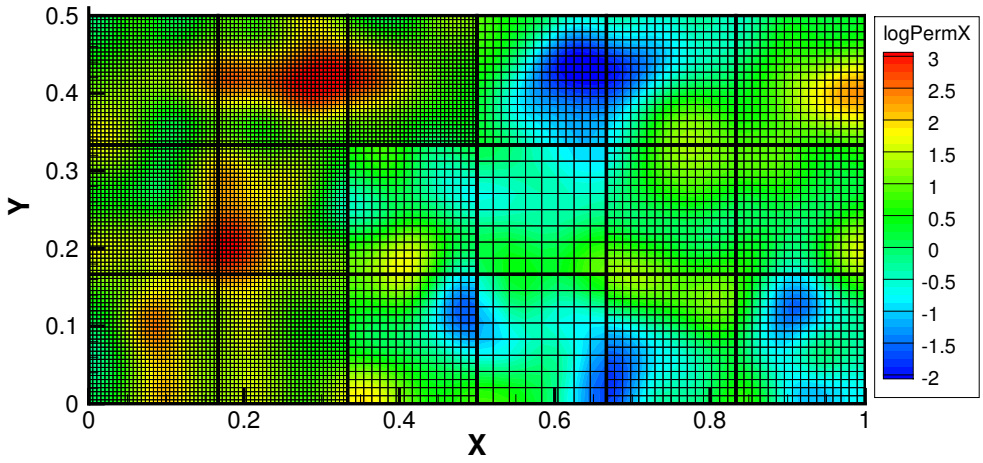

Figure 3.5: Example 3, permeability field on the last mesh refinement level

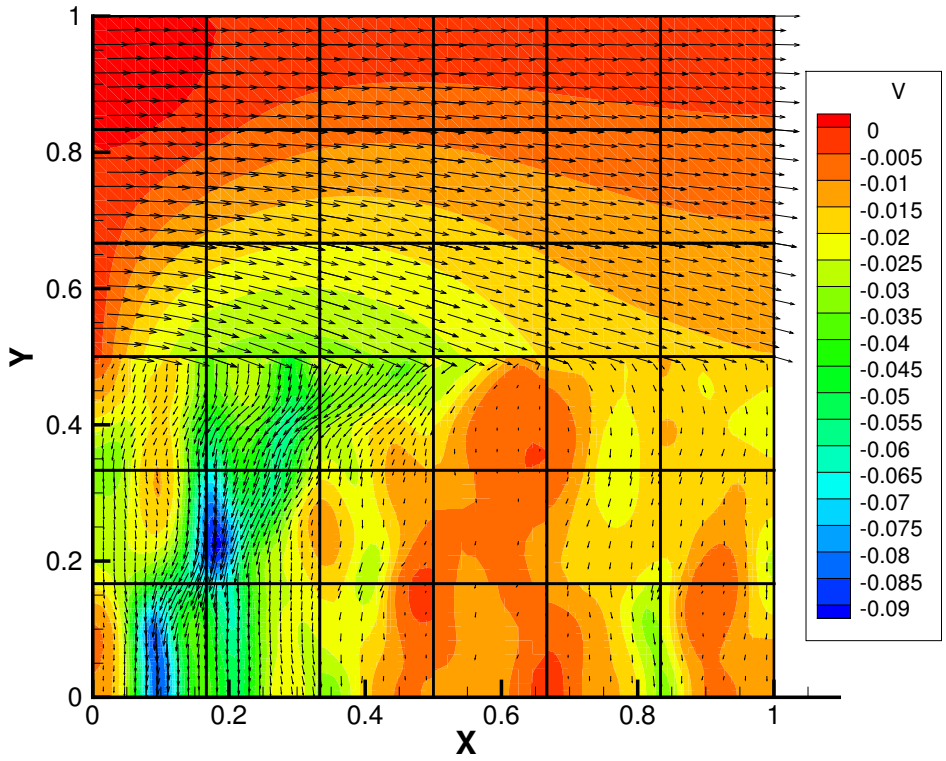

Figure 3.6: Example 3, velocity solution on the last mesh refinement level

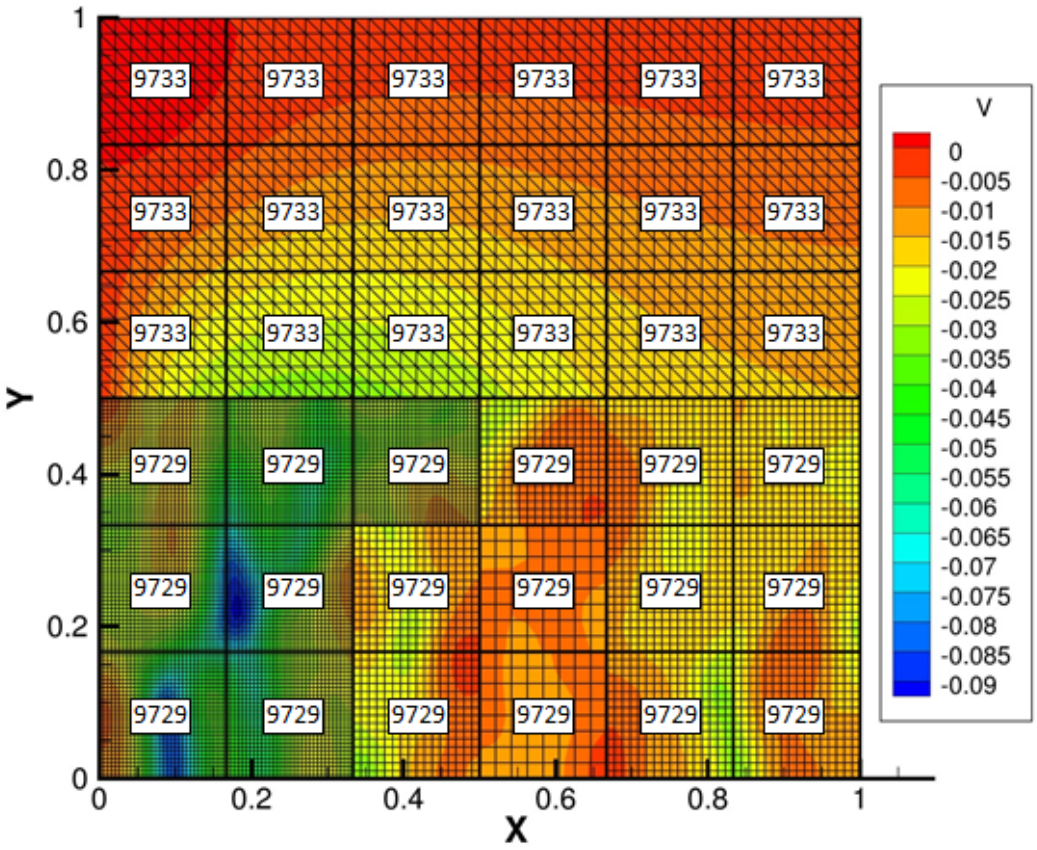

Figure 3.7: Number of solves for Method 1, shown on each subdomain.

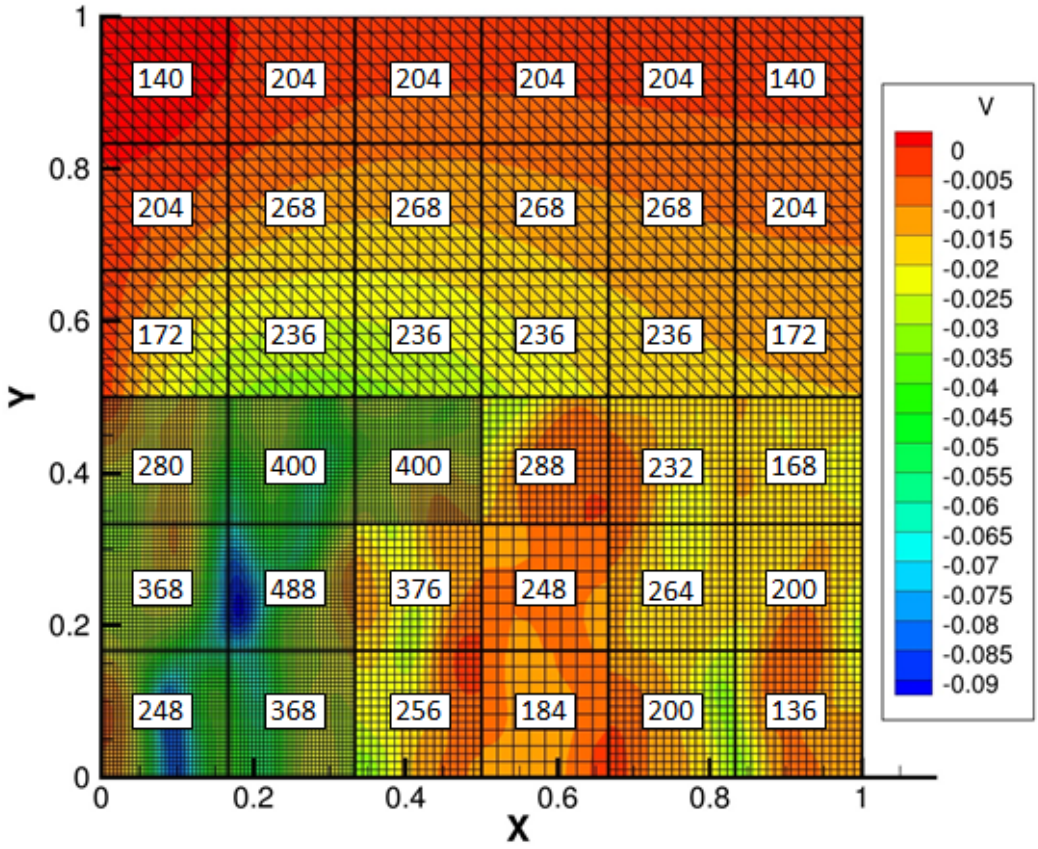

Figure 3.8: Number of solves for Method 3, shown on each subdomain.

4.0 STOCHASTIC MULTISCALE FLUX BASIS FOR STOKES-DARCY FLOWS

The work in this chapter is an application of multiscale flux basis in chapter 3 to stochastic Stokes-Darcy problem, which is to be published in [2]. Physical phenomena arising in the groundwater flow often requires uncertainty quantification to deal with nature of physical parameters used in the model. This is due to the incomplete knowledge of properties of the region. Therefore, we also extend our work to coupled Stokes and Darcy flows with stochastic permeability in this chapter, with its log represented as a sum of local Karhunen-Loève (KL) expansions. Our work is built upon the framework for stochastic collocation and mixed finite elements that was developed in [29]. The problem is approximated by stochastic collocation on either a tensor product or a sparse grid, coupled with multiscale mortar mixed finite element method using non-overlapping domain decomposition for the spatial discretization. Extended model is suitable for studying the interaction between a free fluid and flow in porous media with multiple rock types. Even though the stochasticity comes only from uncertain nature of permeability in porous region, the resulting solution is stochastic due to the coupling conditions.

For the purpose of this work we were parametrizing the permeability function using truncated KL expansion with independent identically distributed random variables. Provided covariance relationship with empirically determined statistics we are able to compute the eigenvalues and corresponding eigenfunctions that form such series. This approach is commonly used for stochastic permeability as shown in [43, 62, 64, 65] and in particular can be used in the framework for stochastic collocation and mixed finite elements [29]. Following [43, 32] we allow nonstationary porous media with different covariance functions for different parts of the domain, providing the ability to model physical phenomena arising in heteroge-

neous media. For instance, the arrangement of sedimentary rocks in distinct layers motivate the use of aforementioned statistically independent regions - each region corresponds to a particular rock type. Further in the chapter such regions are referred to as KL regions. The covariance between two points that lie in different KL regions is zero, while otherwise it depends on distance between those points.

Uncertainty quantification methods can be classified in three major groups: (1) sampling methods [25], (2) moment/perturbation methods [64] and (3) non-perturbative methods, either based on polynomial chaos expansions [63] or stochastic finite element methods [17, 33]. The best known sampling method is Monte Carlo Simulation (MCS). The major disadvantage of this method is the high computational cost due to the need to generate valid representative statistics from a large number of realizations at random points in the stochastic event space. Another highly used approach is stochastic collocation method, which performs collocaction at specifically chosen points. Therefore this technique requires fewer realization, but provides better accuracy than MCS. In this chapter we focus on the choice of collocation points coming from tensor product of zeros of orthogonal polynomials and sparse grid approximations.

In each stochastic realization we solve the coupled Stokes-Darcy problem using the multiscale mortar mixed finite element method (MMMFEM) introduced in [4, 34, 58]. The traditional MMMFEM decomposes the physical domain into a union of non-overlapping subdomains of Stokes or Darcy type, while any Darcy subdomain is assumed to be contained in only one KL region. Like in the deterministic problem, each subdomain has a discretization in fine scale and the grids are allowed to be non-matching along all subdomain interfaces. A mortar space is introduced on the interfaces and discretized in a coarse scale to weakly impose the continuity of physical solutions. Following the algorithm in [35, 58], the global fine scale problem is reduced to a coarse scale interface problem, which can be solved by Krylov iterative solvers in parallel.

To reduce the computational cost in the interface problem, we present two different implementations of multiscale flux basis for Stokes-Darcy flow. Note that in every interface iteration, a Dirichlet solve in each Darcy subdomain and a Neumann solve in each Stokes subdomain are recognized as the dominant cost in MMMFEM, which will grow with the condition number of the interface problem. With (deterministic) multiscale flux basis intro-

duced in Chapter 3, such cost will be eliminated by forming a basis consisting of the solution response for each mortar degree of freedom before the iteration begins. Hence, there would be only a fixed amount of cost for the basis and their linear combination can be used to replace the subdomain solves in the Krylov iteration.

Another way of implementation was proposed in [32] named stochastic multiscale flux basis, which is a combination of the stochastic collocation with the deterministic multiscale basis. Unlike the deterministic implementation, we don't recompute the basis at the beginning of every collocation loop. In this method, a full multiscale basis pre-computation is carried out for all local stochastic realizations before the collocation loop begins. These basis are reused during the collocation step for subdomains with the same local stochastic structure.

The structure of the chapter is listed as follows: First, model problem, KL expansion and some key notations are introduced in section 4.1. Then, the traditional MMMFEM for the stochastic Stokes-Darcy is presented in section 4.3. Tensor product and sparse grid stochastic collocations are presented in section 4.5. Three algorithms, including traditional MMMFEM, deterministic and stochastic multiscale basis implementations are presented in section 4.6. In section 4.7, we will employ all three algorithms on several numerical tests to compare the computational efficiencies.

4.1 MODEL PROBLEM AND KL REGIONS

In this chapter, a stochastic space with probability measure P is denoted by Ω. For any random variable $\xi(\omega) : \Omega \to \mathbb{R}$ with probability density function (PDF) $\rho(z)$, its mean or expectation is defined by

$$E[\xi] = \int_{\Omega} \xi(\omega) dP(\omega) = \int_{\mathbb{R}} z\rho(z) dz, \tag{4.1}$$

and its variance is given by

$$var[\xi] = E[\xi^2] - (E[\xi])^2.$$

We denote the fluid region and porous media region by $D_s \subset \mathbb{R}^d$ and $D_d \subset \mathbb{R}^d$, respectively, where $d = 2, 3$. Let Γ_s be the outside boundary of D_s with outward unit normal vector $\mathbf{n}_s$, and Γ_d be the outside boundary of D_d with outward unit normal $\mathbf{n}_d$. The entire physical domain is defined as $D = D_s \cup D_d$, with the Stokes-Darcy interface denoted by $\Gamma_{SD} = D_s \cap D_d$. Let $(\mathbf{u}_s, p_s)$ and $(\mathbf{u}_d, p_d)$ be the velocity and pressure unknowns in Stokes and Darcy regions, respectively. In Stokes region, let ν_s be the viscosity coefficient and define the deformation rate tensor $\mathbf{D}$ and stress tensor $\mathbf{T}$ by

$$\mathbf{D}(\mathbf{u}_s) := \frac{1}{2}(\nabla \mathbf{u}_s + (\nabla \mathbf{u}_s)^T), \quad \mathbf{T}(\mathbf{u}_s, p_s) := -p_s \mathbf{I} + 2\nu_s \mathbf{D}(\mathbf{u}_s).$$

In Darcy region, let ν_d be the viscosity coefficient and $K(\mathbf{x}, \omega)$ be a stochastic function defined on $D \times \Omega$ representing the nonstationary permeability of porous media. We assume K to be uniformly positive definite for P-almost every $\omega \in \Omega$ with components in $L^\infty(D_d)$. The coupled Stokes-Darcy flow satisfies the following equations (4.2)–(4.10): For P-almost every $\omega \in \Omega$, in Stokes region, $(\mathbf{u}_s, p_s)$ satisfy

$$\begin{aligned} -\nabla \cdot \mathbf{T}(\mathbf{u}_s, p_s) \equiv -2\nu_s \nabla \cdot \mathbf{D}(\mathbf{u}_s) + \nabla p_s &= \mathbf{f}_s \quad \text{in } D_s, && (4.2)\\ \nabla \cdot \mathbf{u}_s &= 0 \quad \text{in } D_s, && (4.3)\\ \mathbf{u}_s &= \mathbf{0} \quad \text{on } \Gamma_s, && (4.4) \end{aligned}$$

where $\mathbf{f}_s$ represents the body force. In Darcy region, $(\mathbf{u}_d, p_d)$ satisfy

$$\begin{aligned} \nu_d K(\mathbf{x}, \omega)^{-1} \mathbf{u}_d + \nabla p_d &= \mathbf{f}_d \quad \text{in } D_d, && (4.5)\\ \nabla \cdot \mathbf{u}_d &= q_d \quad \text{in } D_d, && (4.6)\\ \mathbf{u}_d \cdot \mathbf{n}_d &= 0 \quad \text{on } \Gamma_d, && (4.7) \end{aligned}$$

where $\mathbf{f}_d$ is the gravity force and q_d is an external source or sink term satisfying the solvability condition

$$\int_{D_d} q_d \, d\mathbf{x} = 0.$$

The two regions are coupled on Γ_{SD} through the same interface conditions as in the deterministic problem:

$$\mathbf{u}_s \cdot \mathbf{n}_s + \mathbf{u}_d \cdot \mathbf{n}_d = 0 \quad \text{on } \Gamma_{SD}, \tag{4.8}$$

$$-(\mathbf{T}(\mathbf{u}_s, p_s)\mathbf{n}_s) \cdot \mathbf{n}_s \equiv p_s - 2\nu_s(\mathbf{D}(\mathbf{u}_s)\mathbf{n}_s) \cdot \mathbf{n}_s = p_d \quad \text{on } \Gamma_{SD}, \tag{4.9}$$

$$-\frac{\sqrt{K_j}}{\nu_s \alpha}(\mathbf{T}(\mathbf{u}_s, p_s)\mathbf{n}_s) \cdot \boldsymbol{\tau}_j \equiv -\frac{\sqrt{K_j}}{\alpha} 2(\mathbf{D}(\mathbf{u}_s)\mathbf{n}_s) \cdot \boldsymbol{\tau}_j = \mathbf{u}_s \cdot \boldsymbol{\tau}_j,$$
$$1 \leq j \leq d-1 \qquad \text{on } \Gamma_{SD}. \tag{4.10}$$

On handling the stochastic parameter, we let $Y(\mathbf{x}, \omega) = \ln(K(\mathbf{x}, \omega))$ be the log permeability and define

$$Y'(\mathbf{x}, \omega) := Y(\mathbf{x}, \omega) - E[Y](x).$$

To characterize Y', we divide the Darcy region D_d into N_Ω non-overlapping KL regions $D_d = \bigcup_{i=1}^{N_\Omega} D_{KL}^{(i)}$ with $N_\Omega \geq 1$, where the stochastic structure of every region is independent from the others. In other words, the covariance between any pair of points from different KL regions is zero. The stochastic space is then divided correspondingly by

$$\Omega = \bigotimes_{i=1}^{N_\Omega} \Omega^{(i)}.$$

Therefore, for each event $\omega \in \Omega$, we can write $\omega = (\omega^{(1)}, \ldots, \omega^{(N_\Omega)})$ and

$$Y'(\mathbf{x}, \omega) = \sum_{i=1}^{N_\Omega} Y^{(i)}(\mathbf{x}, \omega^{(i)}).$$

4.2 KARHUNEN-LOÈVE (KL) EXPANSION WITH MULTIPLE KL REGIONS

Each $Y^{(i)}$ has a physical support in $D_{KL}^{(i)}$ and is given a covariance function:

$$C_{Y^{(i)}}(\mathbf{x}, \bar{\mathbf{x}}) = E[Y^{(i)}(\mathbf{x}, \omega^{(i)}) Y^{(i)}(\bar{\mathbf{x}}, \omega^{(i)})].$$

Since it is symmetric and positive definite, it can be decomposed into the following series expansion:

$$C_{Y^{(i)}}(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{j=1}^{\infty} \lambda_j^{(i)} f_j^{(i)}(\mathbf{x}) f_j^{(i)}(\bar{\mathbf{x}}),$$

where the eigenvalues and eigenfunctions $\lambda_j^{(i)}$, $f_j^{(i)}$ respectively, are computed by using $C_{Y^{(i)}}$ as the kernel of Fredholm integral equation:

$$\int_{D_{KL}^{(i)}} C_{Y^{(i)}}(\mathbf{x}, \bar{\mathbf{x}}) f_j^{(i)}(\mathbf{x}) d\mathbf{x} = \lambda_j^{(i)} f_j^{(i)}(\bar{\mathbf{x}}). \tag{4.11}$$

The eigenfunctions of $C_{Y^{(i)}}$ are mutually orthogonal and form a complete spanning set since each $C_{Y^{(i)}}$ is symmetric and positive definite, therefore the Karhunen-Loève expansion for the log permeability can be expressed as

$$Y'(\mathbf{x}, \omega) = \sum_{i=1}^{N_\Omega} \sum_{j=1}^{\infty} \xi_j^{(i)}(\omega^{(i)}) \sqrt{\lambda_j^{(i)}} f_j^{(i)}(\mathbf{x}), \tag{4.12}$$

where the eigenfunctions $f_j^{(i)}(\mathbf{x})$ computed in (4.11) have been extended by zero outside of $D_{KL}^{(i)}$ and the $\xi_j^{(i)} : \Omega_i \to \mathbb{R}$ are independent identically distributed random variables [33]. We would assume for this work that $Y^{(i)}$ are given by Gaussian process, so each $\xi_j^{(i)}$ is a random variable with zero mean and unit variance, with the following probability density function: $\rho_j^{(i)}(y) = 1/\sqrt{2\pi} \exp[-y^2/2]$.

It is reasonable to truncate the KL-expansion after N terms as the eigenvalues $\lambda_j^{(i)}$ show rapid decay when N is large [65]. If the expansion is truncated prematurely, the permeability may appear too smooth in a particular KL region. In our case, for any KL region i, we truncate the expansion after its first $N_{\text{term}}(i)$ terms. Increasing $N_{\text{term}}(i)$ introduces more heterogeneity into the permeability realizations for a chosen region. The modeling error associated with truncating the KL expansion will not be discussed in this thesis, but we

refer the readers to [48] and [14] for some previous related works. The truncation after n_T terms allows us to approximate (4.12) by

$$Y'(\mathbf{x},\omega) \approx \sum_{i=1}^{N_\Omega} \sum_{j=1}^{N_{\text{term}}(i)} \xi_j^{(i)}(\omega^{(i)}) \sqrt{\lambda_j^{(i)}} f_j^{(i)}(\mathbf{x}). \tag{4.13}$$

The above also shows that globally we have $N_{\text{term}} := \sum N_{\text{term}}(i)$ terms in Y'.

The images of the random variables $\mathbb{S}_j^{(i)} = \xi_j^{(i)}(\Omega^{(i)})$ make up finite dimensional spaces which are local to each KL region: $\mathbb{S}^{(i)} = \prod_{j=1}^{N_{\text{term}}(i)} \mathbb{S}_j^{(i)} \subseteq \mathbb{R}^{N_{\text{term}}(i)}$ and also vector space that is global: $\mathbb{S} = \prod_{i=1}^{N_\Omega} \mathbb{S}^{(i)} \subseteq \mathbb{R}^{N_{\text{term}}}$.

Let us introduce a function κ that provides a natural ordering for the global number of stochastic dimensions. Then the j-th stochastic parameter of the i-th KL region have a global index in $\{1,\ldots,N_{\text{term}}\}$ given by the function

$$\kappa(i,j) = \begin{cases} j, & \text{if } i = 1 \\ j + \sum_{k=1}^{i-1} N_{\text{term}}(k), & \text{if } i > 1. \end{cases}$$

Since if $\rho_j^{(i)}$ is the PDF of each $\xi_j^{(i)}$, then joint PDF for ξ is defined to be $\rho = \prod_i \prod_j \rho_j^{(i)}$. These allows us to write $Y(\mathbf{x},\omega) \approx Y(\mathbf{x},\mathbf{y})$, where $\mathbf{y} = \left(\xi_j^{(i)}(\omega^{(i)})\right)_\kappa$.

Note that for the remainder of the chapter, we will simplify the notation by replacing $K(\mathbf{x},\omega)$ with its finite dimensional spectral approximation $K(\mathbf{x},\mathbf{y})$ given by equation (4.13). We also identify each stochastic subspace $\Omega^{(i)}$ with its parameterization $\mathbb{S}^{(i)}$. Therefore the modeling error between the true stochastic solution and its finite dimensional approximation is neglected.

When the exact eigenvalues and eigenfunctions of the covariance function C can be found, the KL expansion is the most efficient method for representing a random field. However, in most cases, closed-form eigenfunctions and eigenvalues are not readily available and numerical procedures need be performed for solving the integral equation (4.11).

4.3 DOMAIN DECOMPOSITION AND VARIATIONAL FORMULATION

Similar to the deterministic case, the coupled Stokes-Darcy flow is solved using domain decomposition method following the approach described in [58]. The Stokes and Darcy domains are partitioned into N_s and N_d non-overlapping subdomains, respectively. Let $N = N_s + N_d$, with $D_s = \bigcup_{i=1}^{N_s} D_i$, $D_d = \bigcup_{i=N_s+1}^{N} D_i$, and $D_i \cap D_j = \emptyset$ for $i \neq j$. Let the interface between adjacent subdomains be $\Gamma_{i,j} = \partial D_i \cap \partial D_j$. Depending on the models of adjacent domains, we group all interfaces into three different types: Stokes-Stokes type, Darcy-Darcy type and Stokes-Darcy type, denoted by Γ_{SS}, Γ_{DD} and Γ_{SD}, respectively. The union of all interfaces is then defined as $\Gamma = \Gamma_{SS} \bigcup \Gamma_{DD} \bigcup \Gamma_{SD}$. In addition, it is assumed that in D_d each KL region is an exact union of subdomains.

Several interface conditions are also applied, we impose continuity of velocity and stress on Γ_{SS}, and continuity of normal velocity and pressure on Γ_{DD}. Let $(\mathbf{u}_i, p_i) = (\mathbf{u}_s|_{D_i}, p_s|_{D_i})$ if D_i is a Stokes subdomain and $(\mathbf{u}_i, p_i) = (\mathbf{u}_d|_{D_i}, p_d|_{D_i})$ if it is a Darcy. Then for P-almost every $\mathbf{y} \in \mathbb{S}$, we seek $(\mathbf{u}_i, p_i)$ for $i = 1, \ldots, N$ that satisfy the equations (1.1)–(1.9) in each subdomain D_i with the following interface conditions:

$$[\mathbf{T}(\mathbf{u}, p)\mathbf{n}] = 0 \quad \text{on } \Gamma_{SS}, \quad [p] = 0 \quad \text{on } \Gamma_{DD}, \tag{4.14}$$

$$[\mathbf{u}] = 0 \quad \text{on } \Gamma_{SS}, \quad [\mathbf{u} \cdot \mathbf{n}] = 0 \quad \text{on } \Gamma_{DD}, \tag{4.15}$$

where $[\,\cdot\,]$ represents the jump on the interface. For example, on Γ_{ij}, $[\mathbf{u}] = (\mathbf{u}_i - \mathbf{u}_j)|_{\Gamma_{ij}}$, $[\mathbf{u} \cdot \mathbf{n}] = \mathbf{u}_i \cdot \mathbf{n}_i + \mathbf{u}_j \cdot \mathbf{n}_j$, using the notation $u_i = u|_{D_i}$, and $\mathbf{n}_i$ is the outer unit normal to ∂D_i.

For clarity, we rename D_i to $D_{s,i}$ if $1 \leq i \leq N_s$, and to $D_{d,i-N_s}$ if $N_s + 1 \leq i \leq N$. In the deterministic setting, the velocity spaces in each domain are defined as

$$\begin{aligned}
\widetilde{V} &= \{\mathbf{v} \in L^2(D)^d;\ \mathbf{v}|_{D_{d,i}} \in \widetilde{V}_D,\ \mathbf{v}|_{D_{s,i}} \in \widetilde{V}_S\},\\
\widetilde{V}_D &= \{\mathbf{v} \in L^2(D_d)^d;\ \mathbf{v}_{d,i} := \mathbf{v}|_{D_{d,i}} \in H(\mathrm{div}; D_{d,i}),\ 1 \leq i \leq N_d,\ \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma_d\},\\
\widetilde{V}_S &= \{\mathbf{v} \in L^2(D_s)^d;\ \mathbf{v}_{s,i} := \mathbf{v}|_{D_{s,i}} \in H^1(D_{s,i})^d,\ 1 \leq i \leq N_s,\ \mathbf{v} = \mathbf{0} \text{ on } \Gamma_s\},
\end{aligned}$$

where

$$H(\mathrm{div}; D_{d,i}) = \{\mathbf{v}_{d,i} \in (L^2(D_{d,i}))^d \mid \nabla \cdot \mathbf{v}_{d,i} \in L^2(D_{d,i})\},$$

equipped with the norm

$$\|\mathbf{v}\|_{H(div;D_{d,i})} = (\|\mathbf{v}\|^2_{L^2(D_{d,i})} + \|\nabla \cdot \mathbf{v}\|^2_{L^2(D_{d,i})})^{1/2}.$$

The pressure space is defined as $\widetilde{W} = L^2_0(D)$, and we also need a space for the Lagrange multiplier to impose continuity on the interfaces with

$$\widetilde{\Lambda} = \widetilde{\Lambda}_{SD} \times \widetilde{\Lambda}_{DD} \times \widetilde{\Lambda}_{SS},$$

$$\widetilde{\Lambda}_{SD} = H^{1/2}(\Gamma_{SD}), \quad \widetilde{\Lambda}_{DD} = \{\mathbf{v} \cdot \mathbf{n};\ \mathbf{v} \in \widetilde{V}_D\} \text{ on } \Gamma_{DD}, \quad \widetilde{\Lambda}_{SS} = \widetilde{V}_S|_{\Gamma_{SS}} \text{ on } \Gamma_{SS},$$

where on Γ_{SS} the Lagrange multiplier has the physical meaning of stress and on $\Gamma_{DD} \bigcup \Gamma_{SD}$ it has the meaning of pressure.

Now we define the following L^2 space on $\mathbb{S}$:

$$L^2_\rho(\mathbb{S}) = \left\{ \mathbf{v} : \mathbb{S} \to \mathbb{R}^d \;\middle|\; \left(\int_{\mathbb{S}} \|\mathbf{v}(\mathbf{y})\|^2 \rho(\mathbf{y}) d\mathbf{y} \right)^{1/2} < \infty \right\},$$

and form the stochastic spaces by taking its tensor product with all the deterministic spaces above:

$$V := V(\mathbb{S}) = \widetilde{V} \otimes L^2_\rho(\mathbb{S}), \quad V_S := V_S(\mathbb{S}) = \widetilde{V}_S \otimes L^2_\rho(\mathbb{S}), \quad V_D := V_D(\mathbb{S}) = \widetilde{V}_D \otimes L^2_\rho(\mathbb{S}),$$

$$W := W(\mathbb{S}) = \widetilde{W} \otimes L^2_\rho(\mathbb{S}), \quad \Lambda := \Lambda(\mathbb{S}) = \widetilde{\Lambda} \otimes L^2_\rho(\mathbb{S}).$$

The weak formulation for the coupled Stokes-Darcy problem (4.2)–(4.10) and (4.14)–(4.15) is given by: Find $(\mathbf{u}, p, \lambda) \in V \times W \times \Lambda$, such that

$$a(\mathbf{u}, \mathbf{v}, \mathbb{S}) + b(\mathbf{v}, p, \mathbb{S}) + b_\Lambda(\mathbf{v}, \lambda, \mathbb{S}) = \int_{\mathbb{S}} \left(\int_D \mathbf{f} \cdot \mathbf{v} \right) \rho(\mathbf{y}) d\mathbf{y}, \quad \forall\, \mathbf{v} \in V, \tag{4.16}$$

$$b(\mathbf{u}, w, \mathbb{S}) = -\int_{\mathbb{S}} \left(\int_{D_d} w\, q_d \right) \rho(\mathbf{y}) d\mathbf{y}, \quad \forall\, w \in W, \tag{4.17}$$

$$b_\Lambda(\mathbf{u}, \mu, \mathbb{S}) = 0, \quad \forall\, \mu \in \Lambda. \tag{4.18}$$

where

$$\tilde{a}_{s,i}(\mathbf{u}_{s,i},\mathbf{v}_{s,i}) = 2\,\nu_s \int_{D_{s,i}} \mathbf{D}(\mathbf{u}_{s,i}) : \mathbf{D}(\mathbf{v}_{s,i}) + \sum_{j=1}^{d-1} \int_{\partial D_{s,i}\cap\Gamma_{sd}} \frac{\nu_s \alpha}{\sqrt{K_j}} (\mathbf{u}_{s,i}\cdot\boldsymbol{\tau}_j)(\mathbf{v}_s\cdot\boldsymbol{\tau}_j),$$

$$1 \le i \le N_s, \quad \forall\, (\mathbf{u}_{s,i},\mathbf{v}_{s,i}) \in V_S \times V_S,$$

$$\tilde{a}_{d,i}(\mathbf{u}_{d,i},\mathbf{v}_{d,i}) = \nu_d \int_{D_{d,i}} K^{-1}\mathbf{u}_{d,i}\cdot\mathbf{v}_{d,i}\ , \quad 1 \le i \le N_d, \quad \forall\, (\mathbf{u}_{d,i},\mathbf{v}_{d,i}) \in V_D \times V_D,$$

$$\tilde{b}_i(\mathbf{v}_i, w_i) = -\int_{D_i} w_i \nabla\cdot\mathbf{v}_i\ , \quad 1 \le i \le N, \quad \forall\, \mathbf{v}_i \in V, \quad \forall\, w_i \in W,$$

$$\tilde{a}(\mathbf{u},\mathbf{v}) = \sum_{i=1}^{N_s} \tilde{a}_{s,i}(\mathbf{u},\mathbf{v}) + \sum_{i=1}^{N_d} \tilde{a}_{d,i}(\mathbf{u},\mathbf{v}), \quad \tilde{b}(\mathbf{v},w) = \sum_{i=1}^{N} \tilde{b}_i(\mathbf{v},w),$$

$$\tilde{b}_\Lambda(\mathbf{v},\mu) = \int_{\Gamma_{SS}} [\mathbf{v}]\,\mu + \int_{\Gamma_{DD}} [\mathbf{v}\cdot\mathbf{n}]\,\mu + \int_{\Gamma_{SD}} [\mathbf{v}\cdot\mathbf{n}]\,\mu, \quad \forall\, (\mathbf{v},\mu) \in V \times \Lambda,$$

$$a(\mathbf{u},\mathbf{v},\mathbb{S}) = \int_{\mathbb{S}} \tilde{a}(\mathbf{u},\mathbf{v})\rho(\mathbf{y})d\mathbf{y}, \quad b(\mathbf{u},\mathbf{v},\mathbb{S}) = \int_{\mathbb{S}} \tilde{b}(\mathbf{u},\mathbf{v})\rho(\mathbf{y})d\mathbf{y}, \quad \forall\, (\mathbf{u},\mathbf{v}) \in V \times V,$$

$$b_\Lambda(\mathbf{v},\mu,\mathbb{S}) = \int_{\mathbb{S}} \tilde{b}_\Lambda(\mathbf{v},\mu)\rho(\mathbf{y})d\mathbf{y}, \quad \forall\, (\mathbf{v},\mu) \in V \times \Lambda.$$

4.4 FINITE ELEMENT DISCRETIZATION

Following the idea in [32], a semidiscrete approximation to the weak solution of the stochastic variational form (4.16) – (4.18) is applied. To do this, we implement the multiscale mortar mixed finite element method (MMMFEM) in the physical dimensions, and then employ the stochastic collocation method, such as tensor product or sparse grid collocation using Gauss-Hermite quadrature rule for the additional stochastic dimensions. Therefore, solving (4.16) – (4.18) is becoming solving a sequence of independent deterministic problem where the original domain decomposition algorithm for Stokes-Darcy coupled problem applies. The resulting solutions are realizations in stochastic space and function values in the quadrature rule.

The MMMFEM we are using allows non-conforming meshes along the subdomain interfaces. Each subdomain D_i is partitioned into a d-dimensional shape regular finite element discretization $\mathcal{T}_{h_i}$ with h_i being the maximal element diameter. Let $\mathcal{T}_h = \bigcup_i \mathcal{T}_{h_i}$ be the global fine mesh and $h = \max_{i=1}^{N} h_i$. On Stokes domains, i.e. $1 \le i \le N_s$, let

$\widetilde{V}_{h,i}(D_i) \times \widetilde{W}_{h,i}(D_i) \subset \widetilde{V}_i(D_i) \times \widetilde{W}_i(D_i)$ be any finite element spaces satisfying the inf-sup condition

$$\inf_{0\neq w_{h,i}\in\widetilde{W}_{h,i}} \sup_{0\neq \mathbf{v}_{h,i}\in\widetilde{V}_{h,i}} \frac{(w_{h,i}, \nabla\cdot \mathbf{v}_{h,i})_{D_i}}{\|\mathbf{v}_{h,i}\|_{H^1(D_i)} \ \|w_{h,i}\|_{L^2(D_i)}} \geq \beta_s > 0. \tag{4.19}$$

where $\widetilde{V}_i(D_i) = \widetilde{V}_S|_{D_i}$ and $\widetilde{W}_i(D_i) = \widetilde{W}|_{D_i}$. Similarly, on Darcy domains, i.e. $N_s+1 \leq i \leq N$, we let $\widetilde{V}_{h,i}(D_i) \times \widetilde{W}_{h,i}(D_i) \subset \widetilde{V}_i(D_i) \times \widetilde{W}_i(D_i)$ be a mixed finite element space on $\mathcal{T}_{h,i}$. Any of the well-known pairs would work include Raviart-Thomas spaces [52], the BDM spaces [11], etc. The global discrete velocity and pressure spaces are given as $\widetilde{V}_h = \bigoplus_{i=1}^N \widetilde{V}_{h,i}$ and $\widetilde{W}_h = \bigoplus_{i=1}^N \widetilde{W}_{h,i}$, respectively.

Next, each subdomain interface $\Gamma_{i,j}$ is partitioned into a coarse $(d-1)$-dimensional quasi-uniform affine mesh $\mathcal{T}_{H_{i,j}}$ with maximal element diameter $H_{i,j}$. A mortar space $\widetilde{\Lambda}_{H_{i,j}}(\Gamma_{i,j}) \subset L^2(\Gamma_{i,j})$ is defined to weakly impose the continuity of normal fluxes for the discrete velocity across the non-matching grids. It contains continuous or discontinuous piecewise polynomials. Let $H = \max_{i,j} H_{i,j}$ and $\widetilde{\Lambda}_H = \bigoplus_{i,j} \widetilde{\Lambda}_{H_{i,j}}$. Then the semidiscrete approximation to (4.16) – (4.18) with stochastic MMMFEM is to find

$$\mathbf{u}_h : \mathbb{S} \to \widetilde{V}_h, \quad p_h : \mathbb{S} \to \widetilde{W}_h, \quad \lambda_H : \mathbb{S} \to \widetilde{\Lambda}_H$$

such that for ρ-almost every $\mathbf{y} \in \mathbb{S}$, the following deterministic problem holds:

$$\tilde{a}(\mathbf{u}_h, \mathbf{v}_h) + \tilde{b}(\mathbf{v}_h, p_h) + \tilde{b}_\Lambda(\mathbf{v}_h, \lambda_H) = \int_D \mathbf{f}\cdot\mathbf{v}_h, \quad \forall\, \mathbf{v}_h \in \widetilde{V}_h, \tag{4.20}$$

$$\tilde{b}(\mathbf{u}_h, w_h) = -\int_{D_d} w_h\, q_d, \quad \forall\, w_h \in \widetilde{W}_h, \tag{4.21}$$

$$\tilde{b}_\Lambda(\mathbf{u}_h, \mu) = 0, \quad \forall\, \mu \in \widetilde{\Lambda}_H. \tag{4.22}$$

The function in mortar space represents the stress on Γ_{SS} and pressure on Γ_{DD}. Therefore the interface condition (4.14) is satisfied on the mortar mesh. On the other hand, condition (4.15), which represents the continuity of velocity on Γ_{SS} and normal velocity on Γ_{DD}, is fulfilled weakly by (4.22).

For well-posedness of the above deterministic problem, refer to [40].

4.4.1 Reduction to interface problem

We use the same algorithm introduced in [58] to solve the deterministic problem (4.20) – (4.22) by reducing them to a symmetric and positive definite interface problem. On each subdomain we split (4.20) – (4.22) into two subproblems. For Stokes domains, $D_{s,i}$, $1 \leq i \leq N_s$, one of the subproblems is to find

$$\mathbf{u}^*_{h,i}(\boldsymbol{\lambda}) : \mathbb{S} \to \widetilde{V}_{h,i}, \quad p^*_{h,i}(\boldsymbol{\lambda}) : \mathbb{S} \to \widetilde{W}_{h,i},$$

with specified $\boldsymbol{\lambda} = (\lambda_n, \boldsymbol{\lambda}_\tau)$, where λ_n and $\boldsymbol{\lambda}_\tau = (\lambda^1_\tau, \ldots, \lambda^{d-1}_\tau)$ represent the normal stress and tangential stress on Γ_{SS}, respectively, such that for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\tilde{a}_{s,i}(\mathbf{u}^*_{h,i}(\boldsymbol{\lambda}), \mathbf{v}_i) + \tilde{b}_i(\mathbf{v}_i, p^*_{h,i}(\boldsymbol{\lambda})) = -\langle \lambda_n, \mathbf{v}_i \cdot \mathbf{n}_i \rangle_{\partial D_{s,i} \setminus \partial D} - \sum_{l=1}^{d-1} \langle \lambda^l_\tau, \mathbf{v}_i \cdot \boldsymbol{\tau}^l_i \rangle_{\partial D_{s,i} \cap \Gamma_{SS}},$$
$$\forall \mathbf{v}_i \in \widetilde{V}_{h,i} / \ker\, \tilde{a}_i, \quad (4.23)$$

$$\tilde{b}_i(\mathbf{u}^*_{h,i}(\boldsymbol{\lambda}), w_i) = 0, \quad \forall\, w_i \in \widetilde{W}_{h,i}, \quad (4.24)$$

where $\{\boldsymbol{\tau}^l_i\}_{l=1}^{d-1}$ is an orthogonal set of unit vectors tangential to $\partial D_{s,i}$ and the kernel space $\ker\, \tilde{a}_i := \{\mathbf{v} \in \widetilde{V}_i : \tilde{a}_i(\mathbf{v}, \mathbf{v}) = 0\}$ consists of a subset of all rigid body motions depending on the boundary types of $D_{s,i}$.

The complementary subproblem is to find

$$\bar{\mathbf{u}}_{h,i} : \mathbb{S} \to \widetilde{V}_{h,i}, \quad \bar{p}_{h,i} : \mathbb{S} \to \widetilde{W}_{h,i},$$

such that for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\tilde{a}_{s,i}(\bar{\mathbf{u}}_{h,i}, \mathbf{v}_i) + \tilde{b}_i(\mathbf{v}_i, \bar{p}_{h,i}) = \int_{D_{s,i}} \mathbf{f}_i \cdot \mathbf{v}_i, \quad \forall \mathbf{v}_i \in \widetilde{V}_{h,i} / \ker\, a_i, \quad (4.25)$$

$$\tilde{b}_i(\bar{\mathbf{u}}_{h,i}, w_i) = 0, \quad \forall w_i \in \widetilde{W}_{h,i}. \quad (4.26)$$

Notice the first problem (4.23) – (4.24) has specified stress on the interface with zero boundary condition and source, while the second problem (4.25) – (4.26) has specified boundary condition and source but with zero stress on the interface. Similarly, we can do the same splitting on Darcy domains $D_{d,i}$, $1 \leq i \leq N_d$, one subproblem is to find

$$\mathbf{u}^*_{h,i}(\lambda) : \mathbb{S} \to \widetilde{V}_{h,i}, \quad p^*_{h,i}(\lambda) : \mathbb{S} \to \widetilde{W}_{h,i},$$

with specified pressure λ such that for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\tilde{a}_{d,i}(\mathbf{u}^*_{h,i}(\lambda), \mathbf{v}_i) + \tilde{b}_i(\mathbf{v}_i, p^*_{h,i}(\lambda)) = -\langle \lambda, \mathbf{v}_i \cdot \mathbf{n}_i \rangle_{\partial D_{d,i} \setminus \partial D}, \quad \forall \mathbf{v}_i \in \widetilde{V}_{h,i}, \tag{4.27}$$

$$\tilde{b}_i(\mathbf{u}^*_{h,i}(\lambda), w_i) = 0, \quad \forall\, w_i \in \widetilde{W}_{h,i}. \tag{4.28}$$

The complementary problem is to find

$$\bar{\mathbf{u}}_{h,i} :\ \mathbb{S} \to \widetilde{V}_{h,i}, \quad \bar{p}_{h,i} :\ \mathbb{S} \to \widetilde{W}_{h,i},$$

such that for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$\tilde{a}_{d,i}(\bar{\mathbf{u}}_{h,i}, \mathbf{v}_i) + \tilde{b}_i(\mathbf{v}_i, \bar{p}_{h,i}) = \int_{D_{d,i}} \mathbf{f}_i \cdot \mathbf{v}_i, \quad \forall \mathbf{v}_i \in \widetilde{V}_{h,i}, \tag{4.29}$$

$$\tilde{b}_i(\bar{\mathbf{u}}_i, w_i) = -\int_{D_{d,i}} w_i\, q_d, \quad \forall w_i \in \widetilde{W}_{h,i}. \tag{4.30}$$

Therefore by equating the solutions via condition (4.22), the original problem (4.20) – (4.22) is equivalent to this interface problem: finding $\lambda_H : \mathbb{S} \to \widetilde{\Lambda}_H$ such that for for ρ-almost every $\mathbf{y} \in \mathbb{S}$,

$$s_H(\lambda_H, \mu) := -\tilde{b}_\Lambda(\mathbf{u}^*_h(\lambda_H), \mu) = \tilde{b}_\Lambda(\bar{\mathbf{u}}_h, \mu), \quad \mu \in \widetilde{\Lambda}_H. \tag{4.31}$$

Problem (4.31) can be solved by a Krylov space solver. In Section 4.7 we use the Conjugate Gradient (CG) method in all the numerical examples. After solving λ_H, the global velocity $\mathbf{u}_h$ and pressure p_h can be recovered by

$$\mathbf{u}_h = \mathbf{u}^*_h(\lambda_H) + \bar{\mathbf{u}}_h, \quad p_h = p_h(\lambda_H) + \bar{p}_h.$$

4.5 STOCHASTIC COLLOCATION

The stochastic collocation method is to approximate the semidiscrete solution $(\mathbf{u}_h, p_h, \lambda_H)$ by an interpolant $\mathcal{I}_m$, where m (or $\mathbf{m}$) is a multi-index indicating the desired polynomial degree of accuracy in stochastic dimensions. It is uniquely formed on a set of N_{real} stochastic points $\mathbf{y}_k$, where N_{real} is a function of m and the fully discrete solution is given by

$$\mathbf{u}_{h,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m \mathbf{u}_h(\mathbf{x},\mathbf{y}), \quad p_{h,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m p_h(\mathbf{x},\mathbf{y}), \quad \lambda_{H,m}(\mathbf{x},\mathbf{y}) = \mathcal{I}_m \lambda_H(\mathbf{x},\mathbf{y}).$$

Choose the Lagrange basis $\{L_m^{\{k\}}(\mathbf{y})\}$ such that $\{L_m^{\{k\}}(\mathbf{y}_j)\} = \delta_{kj}$. Then the Lagrange representation for the fully discrete solution is

$$(\mathbf{u}_{h,m}, p_{h,m}, \lambda_{H,m})(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{N_{\text{real}}} (\mathbf{u}_h^{\{k\}}, p_h^{\{k\}}, \lambda_H^{\{k\}})(\mathbf{x}) L_m^{\{k\}}(\mathbf{y}),$$

where $(\mathbf{u}_h^{\{k\}}, p_h^{\{k\}}, \lambda_H^{\{k\}})$ is the evaluation of semidiscrete solution $(\mathbf{u}_h, p_h, \lambda_H)$ at the point in stochastic space $\mathbf{y}_k$. In other words, for each permeability realization $K^{\{k\}}(\mathbf{x}) = K(\mathbf{x}, \mathbf{y}_k)$, $k = 1, \ldots, N_{\text{real}}$, we solve the deterministic problem: find $\mathbf{u}_h^{\{k\}} \in \widetilde{V}_h$, $p_h^{\{k\}} \in \widetilde{W}_h$ and $\lambda_H^{\{k\}} \in \widetilde{\Lambda}_H$, such that,

$$\tilde{a}(\mathbf{u}_h^{\{k\}}, \mathbf{v}_h) + \tilde{b}(\mathbf{v}_h, p_h^{\{k\}}) + \tilde{b}_\Lambda(\mathbf{v}_h, \lambda_H^{\{k\}}) = \int_D \mathbf{f} \cdot \mathbf{v}_h, \quad \forall\, \mathbf{v}_h \in \widetilde{V}_h, \tag{4.32}$$

$$\tilde{b}(\mathbf{u}_h^{\{k\}}, w_h) = -\int_{D_d} w_h\, q_d, \quad \forall\, w_h \in \widetilde{W}_h, \tag{4.33}$$

$$\tilde{b}_\Lambda(\mathbf{u}_h^{\{k\}}, \mu) = 0, \quad \forall\, \mu \in \widetilde{\Lambda}_H. \tag{4.34}$$

Similarly, equations (4.32)–(4.34) can be reduced to the interface problem: find $\lambda_H^{\{k\}} \in \widetilde{\Lambda}_H$ such that

$$s_H(\lambda_H^{\{k\}}, \mu) := -\tilde{b}_\Lambda(\mathbf{u}_h^{*\{k\}}(\lambda_H^{\{k\}}), \mu) = \tilde{b}_\Lambda(\bar{\mathbf{u}}_h^{\{k\}}, \mu), \quad \mu \in \widetilde{\Lambda}_H, \tag{4.35}$$

using the same technique in section 4.4.1 to split the solutions into "bar" and "star" components.

To get the quadrature rule, one may plug the Lagrange representation of the fully discreet solution into the expectation integral (4.1). Different collocation methods could be obtained by choosing different collocation points $\mathbf{y}_k$. In this chapter we considered tensor product and sparse grids, both of them are constructed using the one-dimensional rules, where the points in dimension $\mathbb{S}_j^{(i)}$ are the zeros of orthogonal polynomials with respect to the $L^2_\rho(\mathbb{S}_j^{(i)})$-inner-product. Since the random variables used in the chapter are Gaussian random variables, it was natural to choose the zeros of the "probabilist" $N(0,1)$ Hermite polynomials

$$H_m(y) = m! \sum_{k=0}^{[m/2]} (-1)^k \frac{(2y)^{m-2k}}{k!(m-2k)!}.$$

The weights and abscissae can be computed with a symbolic manipulation software package or, alternatively, by using a table of rules for the "physicist" $N(0,1/2)$ Hermite polynomials listed in [1] with proper coefficients.

4.5.1 Collocation on Tensor Product Grids

In tensor product collocation, the polynomial accuracy is prescribed in terms of *component* degree. This makes anisotropic rules to be very easy constructed, but the number of points in tensor product rule grows exponentially as number of dimensions or polynomial accuracy are increased. Thus, they are usually used in problems with relatively low number of stochastic dimensions.

Let $N_{\text{term}}(i)$ be the stochastic dimension in KL region i and N_{term} be the total stochastic dimension. We choose $N_{\text{coll}}(i,j)$ many collocation points in stochastic dimension j of KL region i, and define $\mathbf{m} = (N_{\text{coll}}(i,j))_\kappa$, which is an N_{term}-dimensional multi-index, as the required component degree of the interpolant in the stochastic space $\mathbb{S}$. The corresponding anisotropic tensor product Gauss-Hermite interpolant in N_{term}-dimensions is given by

$$\begin{aligned}
\mathcal{I}^{\text{TG}}_{\mathbf{m}} f(\mathbf{y}) &= (\mathcal{I}_{\mathbf{m}(1)} \otimes \cdots \otimes \mathcal{I}_{\mathbf{m}(N_{\text{term}})}) f(\mathbf{y}) \\
&= \sum_{k_1=1}^{\mathbf{m}(1)} \cdots \sum_{k_{N_{\text{term}}}=1}^{\mathbf{m}(N_{\text{term}})} f(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})}) L^{k_1}_{\mathbf{m}(1)}(y_1) \cdots L^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})}(y_{N_{\text{term}}}).
\end{aligned}$$

The set of abscissae for this rule is

$$\mathcal{T}(\mathbf{m}) = \bigotimes_{k=1}^{N_{\text{term}}} \mathcal{H}(\mathbf{m}(k)) = \bigotimes_{i=1}^{N_\Omega} \left(\bigotimes_{j=1}^{N_{\text{term}}(i)} \mathcal{H}(N_{\text{coll}}(i,j)) \right), \tag{4.36}$$

which interpolates the semi-discrete solution into the polynomial space $\mathbb{P}_{\mathbf{m}} = \prod_k \mathbb{P}_{\mathbf{m}(k)}$ in the stochastic dimensions. The tensor product weight for the point $(h^{k_1}_{\mathbf{m}(1)}, \ldots, h^{k_{N_{\text{term}}}}_{\mathbf{m}(N_{\text{term}})})$ is given by

$$w(\mathbf{k}) = \prod_{i=1}^{N_{\text{term}}} w^{k_i}_{\mathbf{m}(i)}.$$

In a fixed stochastic dimension, the one dimensional Gauss-Hermite quadrature rules are accurate to degree $2m-1$.

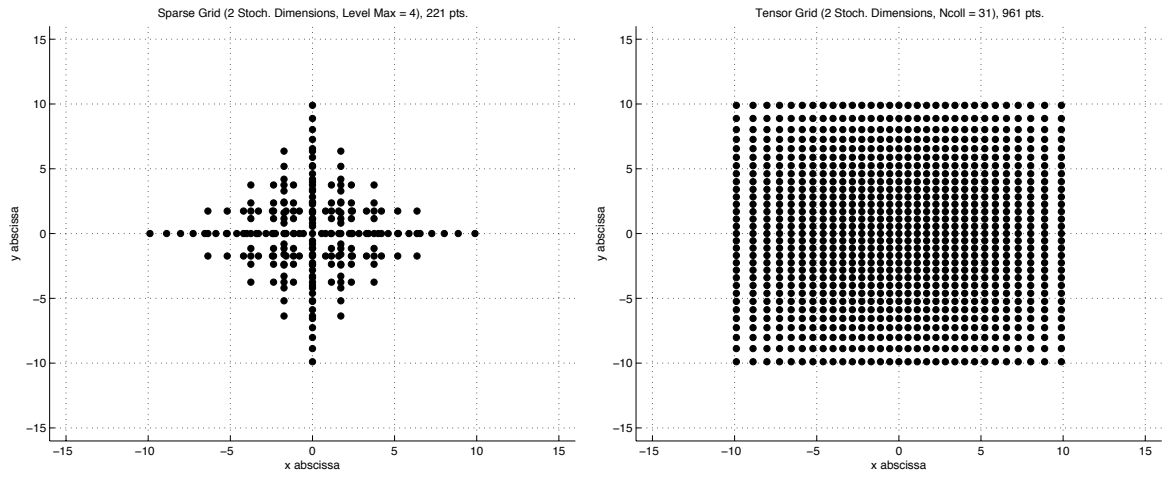

Figure 4.1: *A Gauss-Hermite sparse grid (left) versus a Gauss-Hermite tensor grid (right) with a comparable number of points on each axis.*

4.5.2 Collocation On Sparse Grids

In sparse grid collocation, the polynomial accuracy is prescribed in terms of *total* degree. Sparse grids rules require much fewer points than tensor product rules as the dimension increases, but have the same asymptotic accuracy. This makes them applicable for problems

with high number of stochastic dimensions. A picture of comparable sparse grid and tensor grid rules is shown in Figure 4.1.

Sparse grid rules are linear combination of tensor products on a family of nested one dimensional rules. By the construction sparse grid rules have two main properties: only products with relatively small number of nodes are used and the total polynomial degree is independent of dimension. The characterization of a sparse grid rule is a level $\ell_{\max}$, where the N_{term}-dimensional sparse grid quadrature rule of level $\ell_{\max}$ is accurate to degree $(2 \cdot \ell_{\max} + 1)$.

Each level between $\ell_{\max}$ and $\ell_{\min} = \max\{0, \ell_{\max} - N_{\text{term}} + 1\}$ is an integer split into N_{term} non-negative parts. These partitions defint multi-indices $\mathbf{p} = (p_1, \ldots, p_{N_{\text{term}}})$, where $|\mathbf{p}| = \sum p_i$ denotes the levels of one dimensional rules, which are used for each stochastic dimension. We consider the Gauss-Hermite points $\mathcal{H}(2^{p_i+1} - 1)$ as the one dimensional abscissae of level p. Level 0 consists of a single point, and for every subsequent level the number of points doubles plus one.

For each partition $\mathbf{p}$ consider the multi-index $\mathbf{m} = 2^{\mathbf{p}+\mathbb{1}} - \mathbb{1}$. The corresponding isotropic sparse grid Gauss-Hermite interpolant in N_{term}-dimensions is given by

$$\mathcal{I}^{\text{SG}}_{\ell_{\max}} f(\mathbf{y}) = \sum_{\ell_{\min} \le |\mathbf{p}| \le \ell_{\max}} (-1)^{\ell_{\max} - |\mathbf{p}|} \cdot \binom{N_{\text{term}} - 1}{\ell_{\max} - |\mathbf{p}|} \cdot \mathcal{I}^{\text{TG}}_{\mathbf{m}} f(\mathbf{y}).$$

The set of abscissae for this rule is

$$\mathcal{S}(\ell_{\min}, \ell_{\max}, N_{\text{term}}) = \bigcup_{\ell_{\min} \le |\mathbf{p}| \le \ell_{\max}} \bigotimes_{i=1}^{N_{\text{term}}} \mathcal{H}(2^{p_i+1} - 1). \tag{4.37}$$

Note that the origin is a value which is repeated in each one dimensional rule. Thus, the points in (4.37) are nested weakly.

4.6 COLLOCATION-MMMFEM ALGORITHMS FOR STOKES-DARCY

In this section, we will introduce three different algorithms on the fully discrete stochastic problem (4.32)–(4.34). Recall that on each stochastic realization we solve a deterministic interface problem using CG method which is discretized by MMMFEM. That is, we solve for $\lambda_H^{\{k\}} \in \widetilde{\Lambda}_H$ such that for any $k = 1, \ldots, N_{real}$,

$$s_H^{\{k\}}(\lambda_H^{\{k\}}, \mu) = \tilde{b}_\Lambda(\bar{\mathbf{u}}_h^{\{k\}}, \mu), \quad \forall \mu \in \widetilde{\Lambda}_H. \tag{4.38}$$

Again, for simplicity we introduce the Steklov–Poincaré type operator $S_H^{\{k\}} : \widetilde{\Lambda}_H \to \widetilde{\Lambda}_H$ with

$$\left(S_H^{\{k\}}\lambda, \mu\right) = s_H^{\{k\}}(\lambda, \mu), \quad \forall \lambda, \mu \in \widetilde{\Lambda}_H.$$

Then we can rewrite (4.38) in the operator form

$$S_H^{\{k\}}\lambda_H^{\{k\}} = g_H^{\{k\}}, \tag{4.39}$$

where $g_H^{\{k\}}$ is defined by $\langle g_H^{\{k\}}, \mu\rangle_\Gamma = \tilde{b}_\Lambda(\bar{\mathbf{u}}_h^{\{k\}}, \mu), \quad \forall \mu \in \widetilde{\Lambda}_H$.

It is important to note that the major cost in solving (4.39) comes from the operator action of $S_H^{\{k\}}$ in every CG iteration. This action involves the solving of a Neumann-to-Dirichlet problem (4.23) – (4.24) in Stokes domains, and a Dirichlet-to-Neumann problem (4.27) – (4.28) in Darcy domains.

Three different collocation algorithms are presented below: the first is a traditional M-MMFEM collocation algorithm, the second and the third are implemented with deterministic and stochastic multiscale flux basis, respectively. The second and the third algorithms are dedicated to reduce the number of solves in traditional implementation and achieve a better computational efficiency.

4.6.1 Method S1: Collocation with traditional MMMFEM

Method S1 (without multiscale flux basis)

For $k = 1, \ldots, N_{real}$, do

Step 1: Generate permeability realization $K^{\{k\}}$ corresponding to the global collocation index k.

Step 2: Solve (4.39) using traditional MMMFEM for $\lambda_H^{\{k\}}$.

Step 3: Add the solution to the statistical moments with collocation weight applied: $\lambda_H = \lambda_H + \lambda_H^{\{k\}} \cdot w_m^{\{k\}}$.

End do

As mentioned before, in any subdomain, step 2 costs one subdomain solve from applying $S_H^{\{k\}}$ in every CG iteration. Given mortar function λ_H, the action of $S_H^{\{k\}}\lambda_H$ in subdomain D_i include:

1. Project λ_H onto subdomain boundaries: $\lambda_h = \mathcal{L}_{h,i}\lambda_H$, where $\mathcal{L}_{h,i} : L^2(\Gamma_i) \to \widetilde{V}_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i}$ is the L^2-projection operator onto the normal trace of the velocity space on D_i.
2. If D_i is a Darcy domain, solve the subdomain problem (4.27) – (4.28) with Dirichlet data λ_h; If D_i is a Stokes domain, solve the subdomain problem (4.23) – (4.24) with Neumann data λ_h.
3. Project the resulting flux in Darcy or velocity in Stokes back to mortar space and compute the jump across the interfaces.

Let $N_{\text{iter}}(k)$ be the number of CG iterations for kth realization. Then in any subdomain D_i the leading term in the number of solves for method S1 is

$$N_{S1}(i) := \sum_{k=1}^{N_{\text{real}}} N_{\text{iter}}(k),$$

omitting the extra solves in the right hand side and solution recovery.

In this traditional MMMFEM implementation, since the condition number of our deterministic problem increases with the number of subdomains and magnitude of the permeability, the number N_{iter} in CG can sometimes become large and thus method S1 will be very costly.

4.6.2 Method S2: Collocation with deterministic multiscale flux basis

One way to reduce the cost in MMMFEM for Stokes-Darcy flow is to implement multiscale flux basis following the method introduced in [29] and [30]. When solving the deterministic problem (4.39), we compute the subdomain solution before CG starts for every degree of freedom on mortar space and keep all results as a basis of flux responses in Darcy (or velocity responses in Stokes). Then to evaluate the action of $S_H^{\{k\}}$ in every interface iteration, we simply use the linear combination of the basis instead of costing an extra subdomain solve. The algorithm of this method is presented as follow:

Method S2 (with deterministic multiscale flux basis)

For $k = 1, \ldots, N_{real}$, do

- Step 1: Generate permeability realization $K^{\{k\}}$ corresponding to the global collocation index k.
- Step 2: Compute and save the multiscale flux basis $\{\phi_{H,i}^{\{k\}}\}_{i=1}^{N_{\text{dof}}}$
- Step 3: Solve (4.39) for $\lambda_H^{\{k\}}$ using MMMFEM with the computed basis.
- Step 4: Add the solution to the statistical moments with collocation weight applied: $\lambda_H = \lambda_H + \lambda_H^{\{k\}} \cdot w_m^{\{k\}}$.

End do

In step 2, we follow the routines from traditional MMMFEM and apply the action of operator S_H^k on each mortar basis function. The mortar basis functions $\{\xi_{H,i}^{\{k\}}\}_{i=1}^{N_{\text{dof}}} \subset \widetilde{\Lambda}_H$ are defined such that any $\lambda_H^{\{k\}} \in \widetilde{\Lambda}_H$ can be expressed uniquely by their linear combination:

$$\lambda_H^{\{k\}} = \sum_{i=1}^{N_{\text{dof}}} c_i^{\{k\}} \xi_{H,i}^{\{k\}}.$$

Then for any $i = 1 \ldots N_{\text{dof}}$, the multiscale flux basis is computed by

$$\phi_{H,i}^{\{k\}} = S_H^{\{k\}} \xi_{H,i}^{\{k\}}.$$

To evaluate the result of $S_H^{\{k\}} \lambda_H^{\{k\}}$ in step 3, we simply compute it by

$$S_H^{\{k\}} \lambda_H^{\{k\}} = \sum_{i=1}^{N_{\text{dof}}} c_i^{\{k\}} \phi_{H,i}^{\{k\}}.$$

Since there is no solve needed inside the CG loop, total solves for method S2 would not depend on the number of CG iterations. Therefore, in every stochastic realization, method S2 costs less than method S1 if the number of CG iterations $N_{\text{iter}}(k)$ is bigger than the maximum number of mortar degrees of freedom among all subdomains. In any subdomain D_i, the number of solves for method S2 has the leading term

$$N_{S2}(i) := N_{\text{dof}}(i) \cdot N_{\text{real}}.$$

4.6.3 Method S3: Collocation with stochastic multiscale flux basis

The third algorithm (method S3) was first presented in [32] with an implementation of stochastic multiscale flux basis. This method can achieve greater computational savings than method S2 due to the fact that in the same KL region, both tensor product and sparse grid collocations have a repeated local structure. The main idea is to form a pre-computation loop before the collocation loop, where multiscale flux basis are computed and stored for all local realizations of a subdomain's KL region. Below we present the algorithm of method S3:

Method S3 (with stochastic multiscale flux basis)

For any subdomain D_i, if it is Darcy, then $N_{\text{real}}(j)$ is the number of local realization in the KL region j where D_i belongs to; if it is Stokes, then $N_{\text{real}}(j)$ is set to 1.

Pre-comuputation loop

For $k = 1, \ldots, N_{\text{real}}(j)$, do

Step 1: Generate permeability realization $K^{\{k\}}$ corresponding to the local collocation index k.

Step 2: Compute and save the multiscale flux basis under current local index.

End do

Main loop

For $l = 1, \ldots, N_{real}$, do

Step 3: Generate permeability realization $K^{\{l\}}$ corresponding to the global collocation index l.

Step 4: Convert the gloabal index to the subdomain's local index k

Step 5: Solve (4.39) for $\lambda_H^{\{k\}}$ using MMMFEM with the computed basis for local index k in step 2.

Step 6: Add the solution to the statistical moments with collocation weight applied: $\lambda_H = \lambda_H + \lambda_H^{\{k\}} \cdot w_m^{\{k\}}$.

End do

The algorithms for index conversion for both tensor product grid and sparse grid in step 4 are given in [32], which are numbered as algorithm 5 and 6, respectively. Notice that all subdomain solves (except the extra two for RHS computation and solution recovery) in method S3 are done in the pre-computation loop only. Therefore in subdomain D_i, the total number of solves has the lead term

$$N_{S3}(i) := N_{\text{dof}}(i) \cdot N_{\text{real}}(j).$$

Compared to the leading term in N_{S2} for method S2, N_{S3} is proportional to $N_{\text{real}}(j)$ instead of N_{real}. When there exists more than one KL region in the problem setting, in any KL region j the local realization $N_{\text{real}}(j)$ will be less than the global realization N_{real}, which makes method S3 cost less in subdomain solves than method S2.

4.7 NUMERICAL EXAMPLES

In this section, three numerical examples are presented to test both tensor product and sparse grid collocations and on all methods (S1, S2 and S3). For all tests, we use Taylor-Hood triangular finite element in Stokes and lowest order Raviart-Thomas rectangular finite element in the Darcy for space discretization. All interfaces are discretized via discontinuous piecewise linear mortar finite element.

4.7.1 Test case 1

The first test case has a global domain $[0,1] \times [0,1.2]$, where $D_d = [0,1] \times [0,0.8]$ and $D_s = [0,1] \times [0.8,1.2]$. The problem is divided into 6 equal-size subdomains with two Stokes and four Darcy domains. The outside boundary conditions are given as the following: in Darcy, zero pressure is specified on the bottom edge, and no-flow condition on the left and the right; in Stokes, velocity is specified on the left and top edges, while normal and tangential stresses are specified on the right.

There are two rectangular KL regions in Darcy, defined as $[0,1] \times [0,0.4]$ and $[0,1] \times [0.4,0.8]$, with $N_{\text{term}}(1) = 2 \times 1$ and $N_{\text{term}}(2) = 3 \times 3$ for tensor product grid, $N_{\text{term}}(1) = N_{\text{term}}(2) = 5 \times 5$ for sparse grid. The mean values of both KL regions are read-in from a file with the variances equal to 1.0 and the correlation lengths equal to 0.1. In tensor product collocation, the grid is isotropic and $N_{\text{coll}} = 2$; in sparse grid collocation, $\ell_{\max} = 1$, so both ways have a third degree of accuracy in stochastic dimension. The number of global realization is $N_{\text{real}} = 2048$ for tensor product and $N_{\text{real}} = 201$ for sparse grid.

For space discretization, we have a local mesh of 64×64 in every Darcy domain and 32×32 in every Stokes domain. A mortar mesh of 8×1 is applied to every interface on Γ_{DD} and 4×1 to $\Gamma_{SD} \cup \Gamma_{SS}$.

Table 4.1: *Number of solves and runtime in seconds with the three algorithms for Case 1.*

Tensor Product *Collocation,* $N_{term} = 11$, $N_{coll} = 2$ *(2048 realizations);*

	Method S1	**Method S2**	**Method S3** (Pre-comp)
Max. number of solves	1,499,257	167,936	45,056 (40960)
Runtime in seconds	33457.4	4464.9	3357.1 (23.7)

Sparse Grid *Collocation,* $N_{term} = 50$, $\ell_{max} = 1$ *(101 realizations);*

	Method S1	**Method S2**	**Method S3** (Pre-comp)
Max. number of solves	71,237	8,282	4,282 (4,080)
Runtime in seconds	1566.1	220.5	156.8 (2.5)

Test results for all three methods on tensor product and sparse grid collocations are

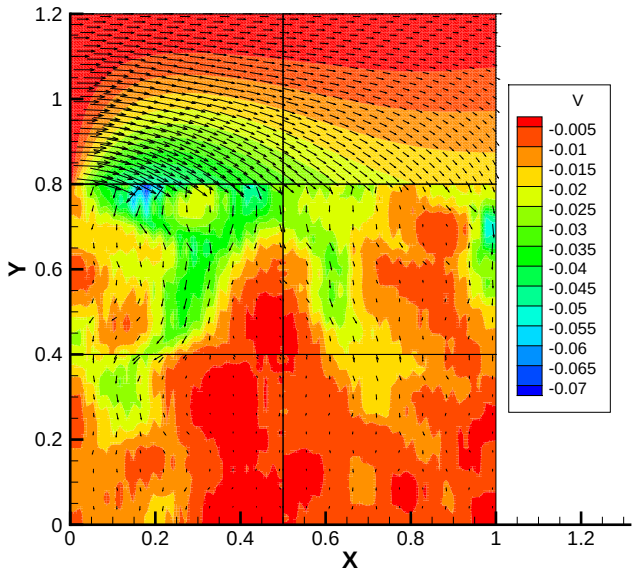

Figure 4.2: *Case 1, realization of solution.*

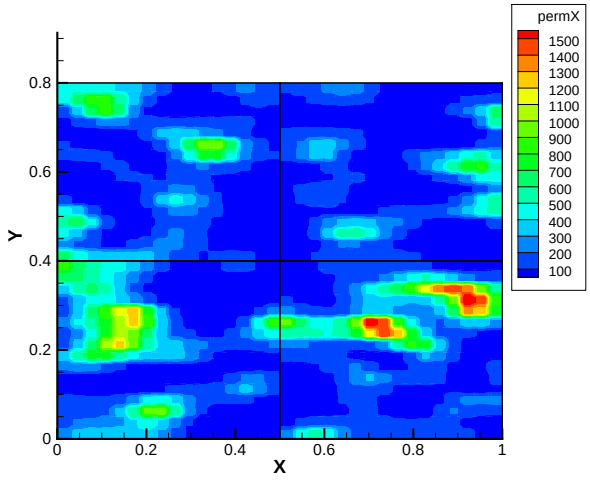

Figure 4.3: *Case 1, realization of permeability.*

shown in Table 4.1. As seen from Figure 4.3, the maximum permeability in this case is of size 10^3, which leads to a large condition number and the number of iterations N_{iter} in each realization. With tensor product collocation, method S1 requires more than 1 million number of solves which is very expensive due to the high value of N_{real} as well as N_{iter}. Method S2 and S3, on the other hand, only cost about 1/10 or less the number of solves and runtime as in method S1, which illustrates some great computational efficiency with multiscale flux basis implemented. Similar results held in sparse grid collocation as well.

It can be easily concluded from Table 4.1 that method S3 has the best performance among all three algorithms. Also, from the values in parenthesis which indicate the cost of the pre-computation, we note that most subdomain solves are completed in the pre-computation loop as expected, while the vast majority of runtime was spent in the main loop. This is because the adding-up cost of the subdomain communications in every CG iteration is much more significant than the cost of multiscale flux basis computation.

Moreover, comparing between method S2 and S3 on both collocation grids, we can see that stochastic multiscale flux basis implementation saves more solves in tensor product (nearly 75%) than sparse grid (nearly 50%). This is due to the fact that for any KL region j, the global to local ratio $N_{\text{real}} : N_{\text{real}}(j)$ is higher in tensor product than in sparse grid.

4.7.2 Test case 2

In this test, the global domain $[0,1]^2$ is divided in half with one Stokes region on the top and one Darcy region on the bottom, and is partitioned into 32 subdomains. Outside boundary condition is given as the following: In Darcy, zero pressure is specified on the bottom edge, with no-flow condition on the left and right; in Stokes, velocity is specified on the left edge, horizontal velocity and normal stress are specified on the top, normal and tangential stresses are specified on the right.

As shown in Figure 4.4, an L-shape KL region $\mathbb{S}_1$ (red part) for stochastic permeability is inscribed in the Darcy region, with its complement $\mathbb{S}_2$ defined to be the second KL region (grey part). In $\mathbb{S}_1$, the mean value is e, variance is 1.0 and correlation lengths are 0.01. In $\mathbb{S}_2$, the mean value is e^{-1}, variance is 1.0 and correlation lengths are 0.1. For tensor product collocation, the grid is isotropic with $N_{\text{coll}} = 2$ and $N_{\text{term}} = 11$, where $N_{\text{term}}(1) = 3 \times 3$ and $N_{\text{term}}(2) = 2 \times 1$; for sparse grid, we have $\ell_{\max} = 1$ and $N_{\text{term}} = 200$, where $N_{\text{term}}(1) = 14 \times 14$ and $N_{\text{term}}(2) = 2 \times 2$. The number of global realization is $N_{\text{real}} = 2048$ for tensor product and $N_{\text{real}} = 401$ for sparse grid.

The space discretization is set as follows: for tensor product, we have a local mesh of 25×25 in every Darcy domain and 8×8 in every Stokes domain. A mortar mesh of 10×1 is applied to every interface on Γ_{DD} and 4×1 to $\Gamma_{SD} \cup \Gamma_{SS}$; for sparse grid, we have a local mesh of 20×20 in every domain in $\mathbb{S}_1$ and 4×4 elsewhere. A mortar mesh of 10×1 is applied to every interior interface of $\mathbb{S}_1$, 4×1 to every outside boundaries of $\mathbb{S}_1$ and 2×1 elsewhere.

The results of this test are shown in Table 4.2 and Figure 4.5-4.6. When multiscale flux basis implemented, the subdomain that having the maximum number of solves is the one at the corner of L-shape who has the highest mortar degrees of freemdom on its interfaces.

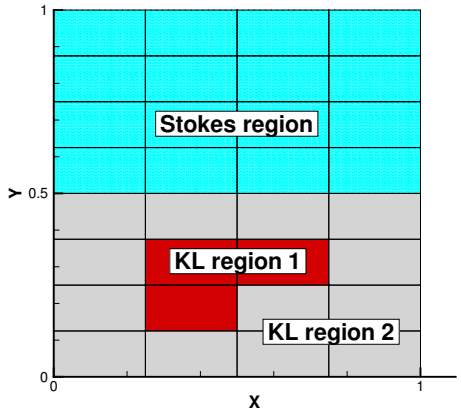

Figure 4.4: *Case2, illustration of different regions.*

Table 4.2: *Number of solves and runtime in seconds with the three algorithms for Case 2.*

Tensor Product *Collocation,* $N_{term} = 11$, $N_{coll} = 2$ *(2048 realizations);*

	Method S1	**Method S2**	**Method S3** (Pre-comp)
Max. number of solves	3,033,494	331,776	86,016 (81,920)
Runtime in seconds	5893.3	3155.9	2707.6 (16.5)

Sparse Grid *Collocation,* $N_{term} = 200$, $\ell_{max} = 1$ *(401 realizations);*

	Method S1	**Method S2**	**Method S3** (Pre-comp)
Max. number of solves	346,709	45,714	44,818 (44,016)
Runtime in seconds	525.5	435.1	391.2 (2.9)

Comparing method S2 with S1, we note that it again saves more than 85% in number of solves, but saves only 50% in time for tensor product and 20% for sparse grid. It is because this test has a much heavier interprocess communication involving 32 cores. The gain in method S3 is moderate for tensor product collocation but a bit limited in sparse grid. Specifically, in tensor product, we reduce the number of solves by 70% and only by 10% in

time compared with method S2; in sparse grid, the save is 2% in solves and 10% in time. Significant communication cost explains the limited gain in time for S3 since in both cases it took very little time to pre-compute all the stochastic multiscale basis.

Another reason that we are not gaining much in method S3, especially in sparse grid, is a very low global to local ratio $N_{\text{real}} : N_{\text{real}}(j)$. In KL region $\mathbb{S}_1$, where $N_{\text{term}}(1) = 196$, the local realization number is $N_{\text{real}}(1) = 393$. Therefore, $N_{\text{real}} : N_{\text{real}}(1)$ equals only about 1.02 in $\mathbb{S}_1$, making the difference very little between $N_{S3}(i)$ and $N_{S2}(i)$ for any $D_i \subset \mathbb{S}_1$.

4.7.3 Test case 3

The third test in this section is a coupled surface water and groundwater flow with realistic geometry. The outside boundary conditions for Darcy region are the same as in test case 1; for Stokes region, velocity is specified on the left and right, normal and tangential stresses are specified on the top. There are $4 \times 2 = 8$ domains on the global irregular region with Stokes on the top half and Darcy on the bottom. The permeability field in one example tensor product realization is presented in Figure 4.7. As shown in Figure 4.8, it is generated from four KL regions, represented by each one of the four Darcy subdomains, respectively. In all KL regions, the mean values of the permeability are read from a file, the variance equals to 1 and correlation lengths are 0.1. For tensor product, the grid is isotropic with $N_{\text{coll}} = 2$, the stochastic dimensions are set by $N_{\text{term}}(j) = 2 \times 1$ for $j = 1, 2, 3, 4$ and $N_{\text{term}} = 8$ in total. For sparse grid, we have $\ell_{\max} = 1$ and $N_{\text{term}} = 100$ with $N_{\text{term}}(j) = 5 \times 5$ for $j = 1, 2, 3, 4$. The physical grids are alternating 18×15 and 15×12 in Darcy domains, and 12×15 and 9×12 in Stokes. A mortar mesh of 4×1 are used on all interfaces.

To handle irregular geometry, we employ the multipoint flux mixed finite element method in the Darcy region [60] which can reduce to a cell-centered finite difference for the pressure. In Stokes region, we impose the mortar conditions on curved interfaces by mapping the physical grids to reference grids with flat interfaces. For more details about this implementation, please refer to [55].

Figure 4.7 – 4.11 show the permeability and solution realization in sparse grid, with solution means and variance in both grids. There is little difference between the computed

means in tensor product and sparse grid since both cases have the same mean permeability sourced from a file. The computed variance is of larger size in sparse grid due to the more collocation points picked for every KL region.

In this problem, a smaller size in stochastic dimension leads to less global realizations in the loop, with $N_{\text{real}} = 256$ in tensor product and $N_{\text{real}} = 201$ in sparse grid. In general, a smaller collocation loop means a fewer saves in aggregate for methods S2 and S3. The computation costs of this test case are presented in Table 4.3. Compared with traditional implementation, Method S2 results in a $50\% \sim 60\%$ decrease in solves and a $30\% \sim 40\%$ decrease in runtime.

With 4 independent KL regions, we have a larger global to local ratio such that for any j, $N_{\text{real}} : N_{\text{real}}(j) = 256 : 4 = 64$ for tensor product and $N_{\text{real}} : N_{\text{real}}(j) = 201 : 51 \approx 3.94$ for sparse grid. That is why Methods S3 becomes significantly better than method S2 with a large gain in numbers of solves and a moderate gain in time, especially for tensor product. Also we note that this is a very cheap improvement providing the low cost of runtime in pre-computation.

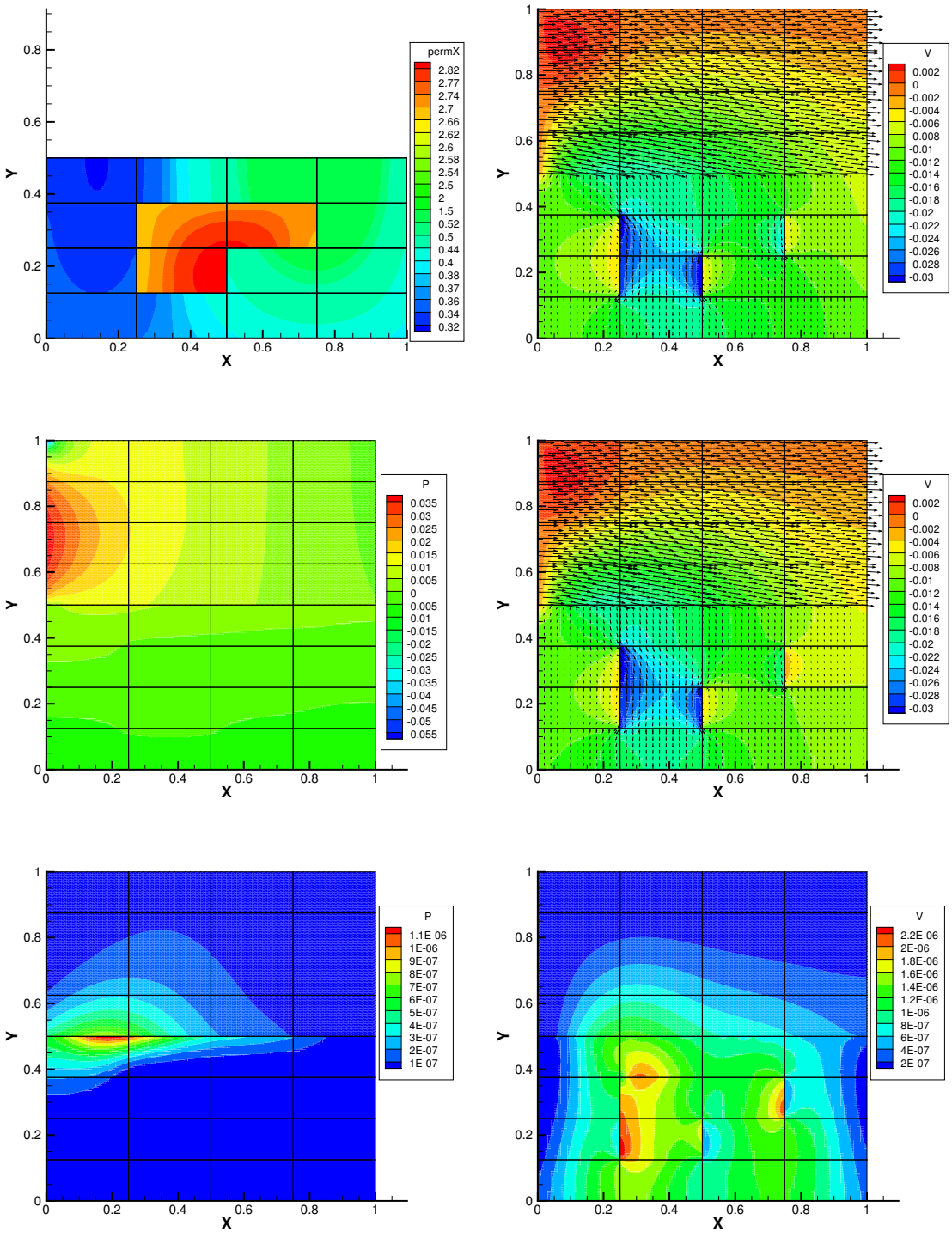

Figure 4.5: *Case 2, tensor product, realization of permeability (top-left), realization of solution (top-right), mean value of pressure (middle-left), mean value of vertical solution (middle-right), variance of pressure (bottom-left), variance of vertical solution (bottom-right)*

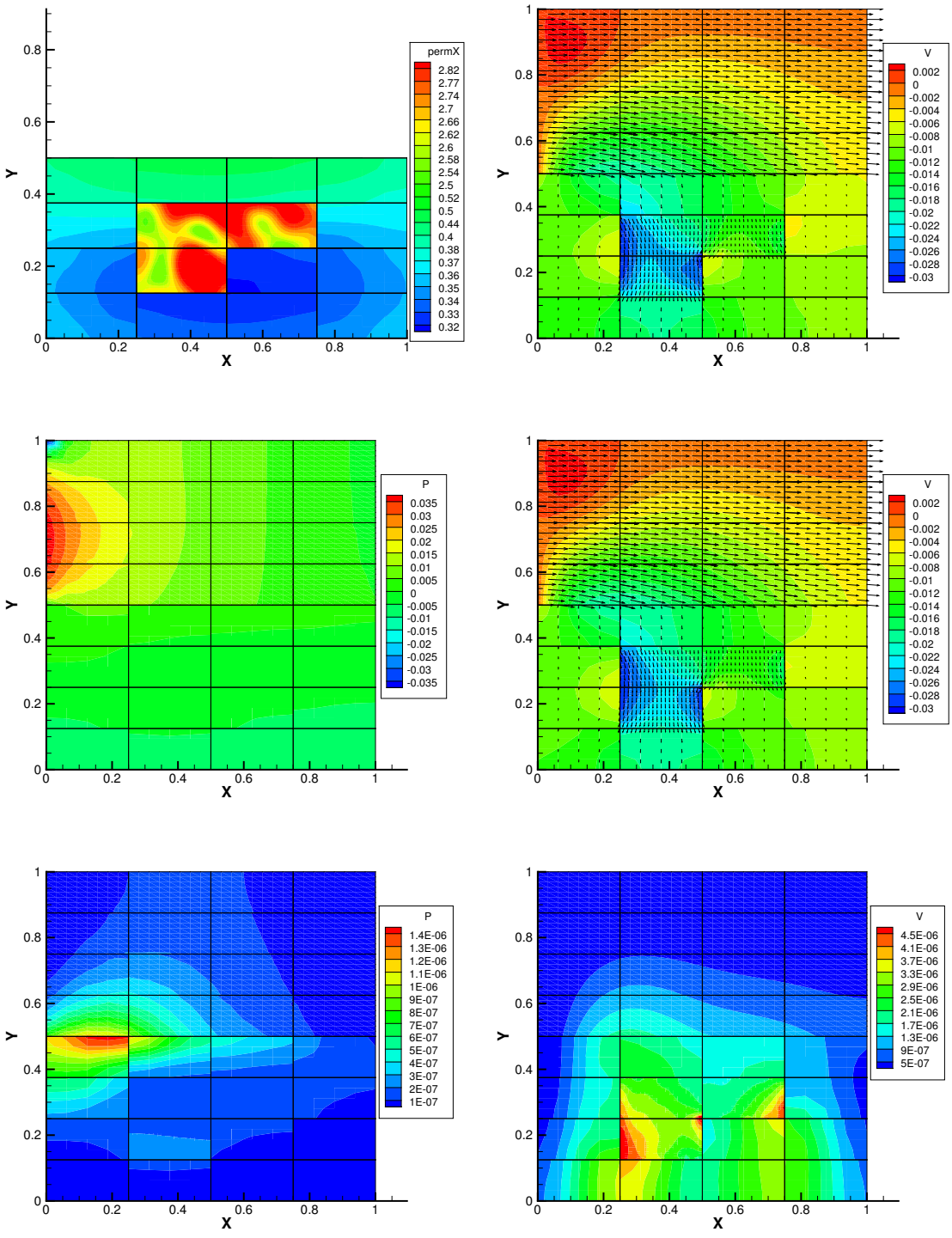

Figure 4.6: *Case 2 sparse grid , realization of permeability (top-left), realization of solution (top-right), mean value of pressure (middle-left), mean value of vertical solution (middle-right), variance of pressure (bottom-left), variance of vertical solution (bottom-right)*

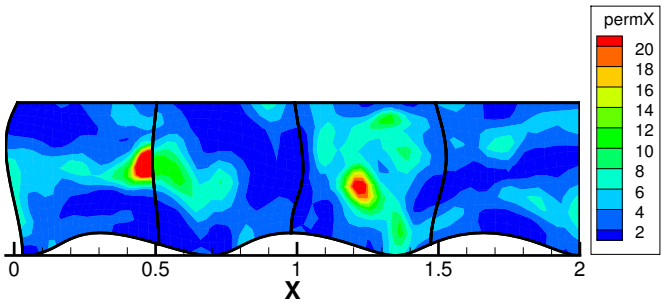

Figure 4.7: *Case 3, realization of permeability.*

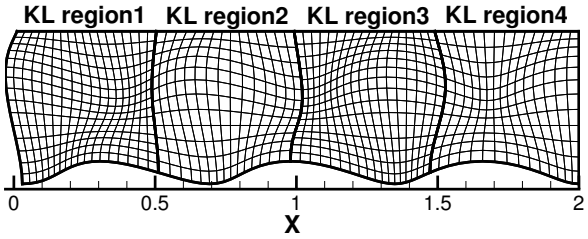

Figure 4.8: *Case 3, KL regions and local meshes.*

Table 4.3: *Number of solves and runtime in seconds with the three algorithms for Case 3.*

Tensor Product *Collocation,* $N_{term} = 8$, $N_{coll} = 2$ *(256 realizations);*

	Method S1	**Method S2**	**Method S3** (Pre-comp)
Max. number of solves	58,728	21,248	960 (192)
Runtime in seconds	255.5	192.8	91.7 (0.3)

Sparse Grid *Collocation,* $N_{term} = 100$, $\ell_{max} = 1$ *(201 realizations);*

	Method S1	**Method S2**	**Method S3** (Pre-comp)
Max. number of solves	45,031	16,683	3,051 (2,448)
Runtime in seconds	178.3	107.4	82.6 (0.5)

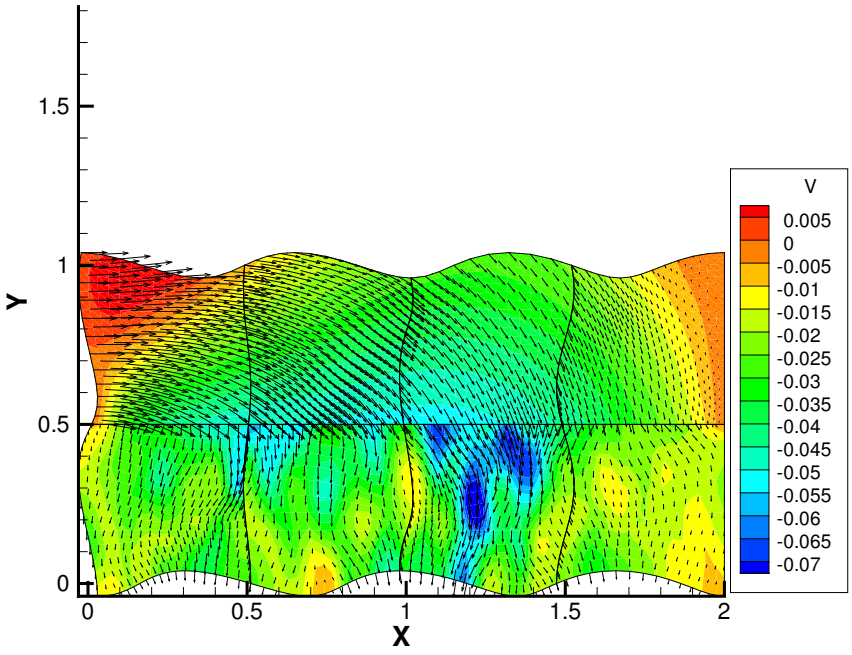

Figure 4.9: *Case3, realization of solution.*

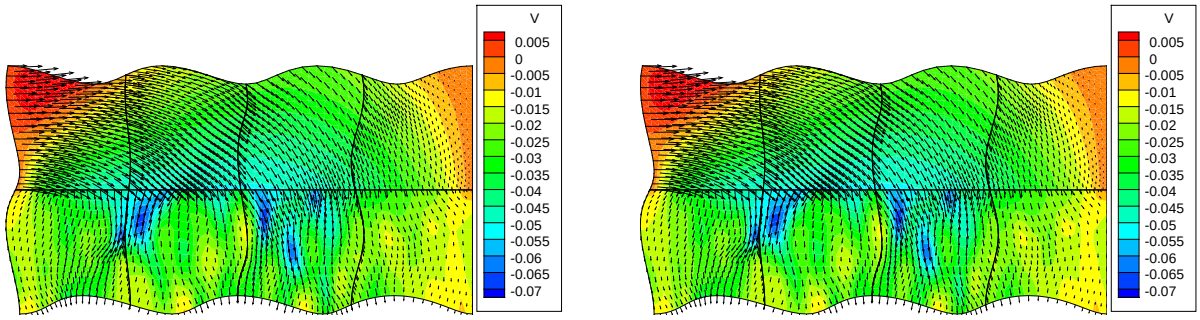

Figure 4.10: *Case3, mean of vertical velocity for tensor product (left) and sparse grid (right).*

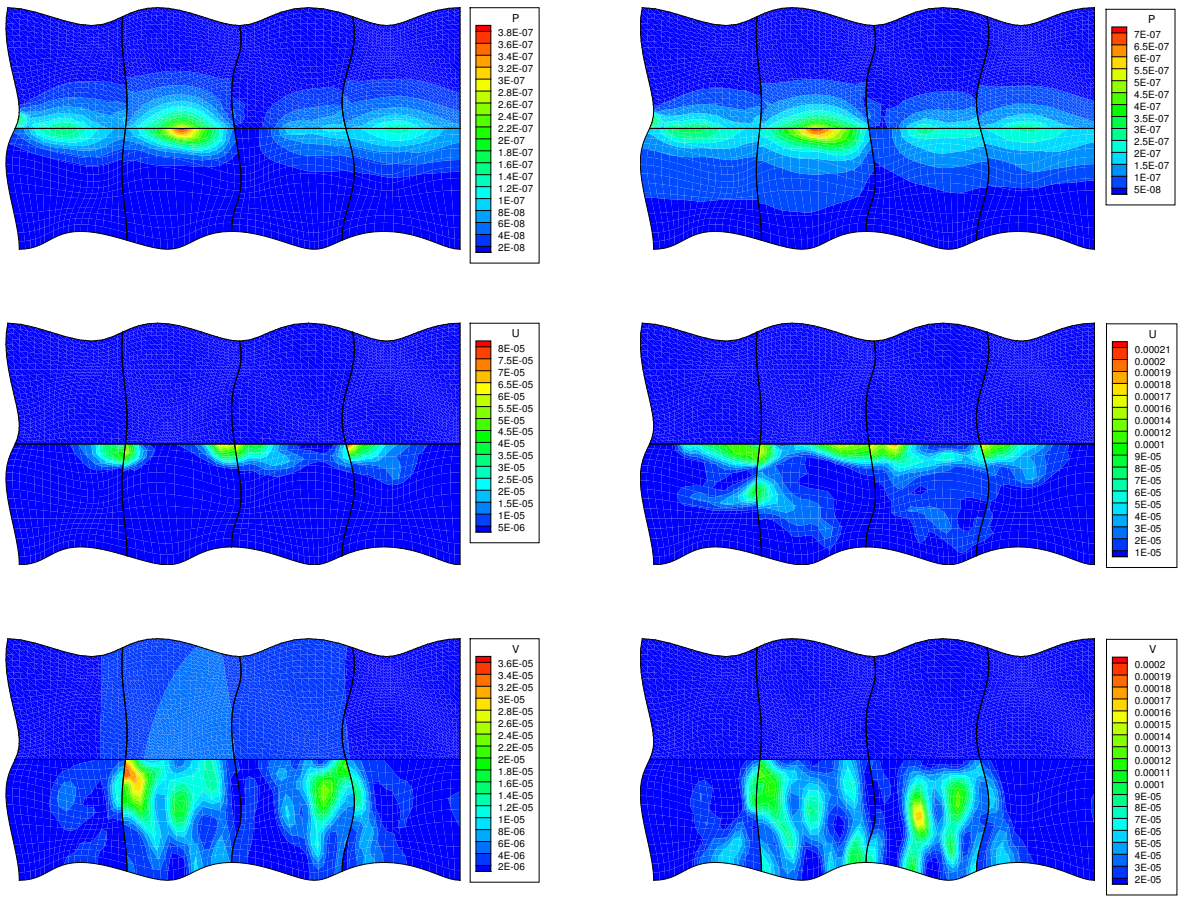

Figure 4.11: *Case 3, solution variance for tensor product (left) and sparse grid (right)*

5.0 CONCLUSION AND FUTURE WORK

In this thesis, we presented several parallel algorithms for coupled Stokes-Darcy flow based on domain decomposition method.

First, the traditional domain decomposition algorithm is introduced, as well as the multiscale mortar mixed finite element method (MMMFEM) which allows non-matching grids in different subdomains. In particular, FETI method is applied to handle singular Stokes subdomains. Condition number estimate of the resulting interface problem for the non-mortar case is shown and then verified in the numerical examples.

Second, we developed a multiscale flux basis algorithm for the MMMFEM discretization and compare it to the original implementation. The new method precomputes the basis by solving a fixed number of subdomain solves, which depend on the number of coarse scale mortar degrees of freedom per subdomain. The basis is used to replace the subdomain solves in the global interface iteration, completely eliminating the dependency between the number of solves and the number of iterations. The numerical examples for a variety of test cases show that the multiscale flux basis implementation is more computationally efficient than the original MMMFEM, and a greater savings occurs with increased number of subdomains and global mortar degrees of freedom. We also demonstrate the multiscale flux basis can be combined with the balancing preconditioner in Darcy to form a powerful solver for large-scale problems.

Third, we presented three algorithms on coupled stochastic Stokes-Darcy problem, with one of which implements no multiscale flux basis and the other two are featuring deterministic and stochastic multiscale flux basis, respectively. Numerical tests in this section demonstrate a significant save in number of solves and runtime with multiscale flux basis implemented on the reduced mortar interface problem of coarse scale. In the cases with large global to

local realizations ratio, such as using tensor product collocation or having more KL regions, stochastic multiscale flux basis stands out for gaining more from the extra pre-computation. In all algorithms, the subdomain communication costs in each CG iteration become a huge factor when the local problems are relatively cheap. A balancing preconditioner for Darcy region [44, 50] could be used to make an improvement by reducing the number of iterations.

As for future work, one interesting topic is to add stochasticity in the source term of Stokes equation, i.e., modeling rainfall in surface-subsurface flow simulations. Similar performances should be expected for all algorithms discussed in Chapter 4.

Another topic would be to extend Theorem 2.3.1 to a more general non-matching grids case and carry out proper analysis of the interface operator and estimate of the condition number.

In addition, to control the minimum eigenvalue of the interface problem 2.20 and achieve a better bounds, FETI preconditioner for Stokes can be implemented. Its operator form writes

$$B_S^{-1} = \sum_{i=1}^{N_s} L A_{h,i}^{+} L^T,$$

where in every iteration a local Dirichlet problem is solved on each Stokes domain [42][50]. The condition number for FETI and Balancing preconditioned system is expected to be similar to $C(1+log(H/h))^2$, and it is natural to expect a faster and more efficient algorithm combining FETI with balancing preconditioner or with multiscale flux basis. The analysis and numerical tests will be carried out in future works.

BIBLIOGRAPHY

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, 9th printing edition, 1972.

[2] I. Ambartsumyan, E. Khattatov, C. Wang, and I. Yotov. Stochastic multiscale flux basis for Stokes-Darcy flows. In preparation.

[3] T. Arbogast. Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems. *SIAM J. Numer. Anal.*, 42:576–598, 2004.

[4] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov. A Multiscale Mortar Mixed Finite Element Method. *Multiscale Model. Simul.*, 6(1):319, 2007.

[5] Todd Arbogast and Dana S. Brunson. A computational method for approximating a darcy–stokes system governing a vuggy porous medium. *Computational Geosciences*, 11(3):207–218, 2007.

[6] D. N. Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344 (1985), 1984.

[7] G. S. Beavers and D. D. Joseph. Boundary conditions at a naturally impermeable wall. *J. Fluid. Mech*, 30:197–207, 1967.

[8] C. Bernardi and G. Raugel. Analysis of some finite elements for the Stokes problem. *Mathematics of Computation*, pages 71–79, 1985.

[9] C. Bernardi, T. C. Rebollo, F. Hecht, and Z. Mghazli. Mortar finite element discretization of a model coupling Darcy and Stokes equations. *M2AN Math. Model. Numer. Anal.*, 42(3):375–410, 2008.

[10] F. Brezzi, J. Douglas, Jr., R. Duràn, and M. Fortin. Mixed finite elements for second order elliptic problems in three variables. *Numer. Math.*, 51:237–250, 1987.

[11] F. Brezzi, J. Douglas, Jr., and L. D. Marini. Two families of mixed elements for second order elliptic problems. *Numer. Math.*, 88:217–235, 1985.

[12] Yanzhao Cao, Max Gunzburger, Xiaoming He, and Xiaoming Wang. Robinrobin domain decomposition methods for the steady-state stokesdarcy system with the beaversjoseph interface condition. *Numerische Mathematik*, 117(4):601–629, 2011.

[13] Yanzhao Cao, Max Gunzburger, Xiaolong Hu, Fei Hua, Xiaoming Wang, and Weidong Zhao. Finite element approximations for stokesdarcy flow with beaversjoseph interface conditions. *SIAM Journal on Numerical Analysis*, 47(6):4239–4256, 2010.

[14] H. Chang and D. Zhang. A comparative study of stochastic collocation methods for flow in spatially correlated random fields. *Commun. Comput. Phys.*, 6(3):509–535, 2009.

[15] Z. Chen and T. Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. of Comp.*, 72:541–576, 2003.

[16] L. C. Cowsar, J. Mandel, and M. F. Wheeler. Balancing domain decomposition for mixed finite elements. *Math. of Comp.*, 64(211):989–1015, 1995.

[17] M. K. Deb, I. M. Babuška, and J. T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Comput. Methods Appl. Mech. Eng.*, 190(48):6359–6372, 2001.

[18] M. Discacciati and A. Quarteroni. Analysis of a domain decomposition method for the coupling of Stokes and Darcy equations. In *Numerical mathematics and advanced applications*, pages 3–20. Springer Italia, Milan, 2003.

[19] Marco Discacciati, Edie Miglio, and Alfio Quarteroni. Mathematical and numerical models for coupling surface and groundwater flows. *Appl. Numer. Math.*, 43(1-2):57–74, 2002. 19th Dundee Biennial Conference on Numerical Analysis (2001).

[20] Marco Discacciati and Alfio Quarteroni. Convergence analysis of a subdomain iterative method for the finite element approximation of the coupling of Stokes and Darcy equations. *Comput. Vis. Sci.*, 6(2-3):93–103, 2004.

[21] Marco Discacciati, Alfio Quarteroni, and Alberto Valli. Robin-Robin domain decomposition methods for the Stokes-Darcy coupling. *SIAM J. Numer. Anal.*, 45(3):1246–1268 (electronic), 2007.

[22] Y. Efendiev, J. Galvis, and T. Y. Hou. Generalized multiscale finite element methods (GMsFEM). *J. Comput. Phys.*, 251:116–135, 2013.

[23] Y. Efendiev, J. Galvis, and X.-H. Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *J. Comput. Phys.*, 230(4):937–955, 2011.

[24] C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Methods Engrg.*, 32:1205–1227, 1991.

[25] G.S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer, Berlin, 1996.

[26] J. Galvis and Y. Efendiev. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Model. Simul.*, 8(5):1621–1644, 2010.

[27] Juan Galvis and Marcus Sarkis. Non-matching mortar discretization analysis for the coupling Stokes-Darcy equations. *Electron. Trans. Numer. Anal.*, 26:350–384, 2007.

[28] Juan Galvis and Marcus Sarkis. FETI and BDD preconditioners for Stokes-Mortar-Darcy systems. *Commun. Appl. Math. Comput. Sci.*, 5:1–30, 2010.

[29] B. Ganis, H. Klie, M.F. Wheeler, T. Wildey, I. Yotov, and D. Zhang. Stochastic collocation and mixed finite elements for flow in porous media. *Comput. Methods Appl. Mech. Eng.*, 197(43-44):3547–3559, 2008.

[30] B. Ganis, D. Vassilev, C. Wang, and I. Yotov. A multiscale flux basis for mortar discretizations of Stokes-Darcy flows. Submitted.

[31] B. Ganis and I. Yotov. Implementation of a Mortar Mixed Finite Element Method using a Multiscale Flux Basis. *Comput. Methods Appl. Mech. Eng.*, 198(49-52):3989–3998, 2009.

[32] B. Ganis, I. Yotov, and M. Zhong. A stochastic mortar mixed finite element method for flow in porous media with multiple rock types. *SIAM Journal on Scientific Computing*, 33(3):1439–1474, 2011.

[33] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, New York, 1991.

[34] V. Girault, D. Vassilev, and I. Yotov. Mortar multiscale finite element methods for Stokes-Darcy flows. *Numerische Mathematik*, 127:93–165, 2014.

[35] R. Glowinski and M. F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA*, 1988.

[36] Ronald H. W. Hoppe, Paulo Porta, and Yuri Vassilevski. Computational issues related to iterative coupling of subsurface and channel flows. *Calcolo*, 44(1):1–20, 2007.

[37] T. Y. Hou and X. H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[38] T. J. R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comput. Methods Appl. Mech. Engrg.*, 127:387–401, 1995.

[39] Trygve Karper, Kent-Andre Mardal, and Ragnar Winther. Unified finite element discretizations of coupled darcystokes flow. *Numerical Methods for Partial Differential Equations*, 25(2):311–326, 2009.

[40] W. J. Layton, F. Schieweck, and I. Yotov. Coupling fluid flow with porous media flow. *SIAM J. Numer. Anal.*, 40(6):2195–2218, 2003.

[41] V. I. Lebedev. *Composition methods*. USSR Academy of Sciences, Moscow, 1986. In Russian.

[42] J. Li. A dual-primal FETI method for incompressible Stokes equations. *Numerische Mathematik*, 102:257–275, 2005.

[43] Z. Lu and D. Zhang. Stochastic simulations for flow in nonstationary randomly heterogeneous porous media using a KL-based moment-equation approach. *Multiscale Model. Simul.*, 6(1):228–245, 2008.

[44] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods for Engineering*, 9(3):233–241, 1993.

[45] J. Mandel and M. Brezina. Balancing domain decomposition for problems with large jumps in coefficients. *Mathematics of Computation*, 65(216):1387–1401, 1996.

[46] Kent Andre Mardal, Xue-Cheng Tai, and Ragnar Winther. A robust finite element method for Darcy-Stokes flow. *SIAM J. Numer. Anal.*, 40(5):1605–1631 (electronic), 2002.

[47] M. Mu and J. Xu. A two-grid method of a mixed stokesdarcy model for coupling fluid flow with porous media flow. *SIAM Journal on Numerical Analysis*, 45(5):1801–1813, 2007.

[48] F. Nobile, R. Tempone, and C.G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2309–2345, 2008.

[49] G. Pacquaut, J. Bruchon, N. Moulin, and S. Drapier. Combining a level-set method and a mixed stabilized p1/p1 formulation for coupling stokesdarcy flows. *International Journal for Numerical Methods in Fluids*, 69(2):459–480, 2012.

[50] G. Pencheva and I. Yotov. Balancing domain decomposition for mortar mixed finite element methods on non-matching grids. *Numer. Linear Algebra Appl.*, 10(1-2):159–180, 2003.

[51] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential equations*. Clarendon Press, Oxford, 1999.

[52] R. A. Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. In *Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics*, volume 606, pages 292–315. Springer-Verlag, New York, 1977.

[53] B. Rivière and I. Yotov. Locally conservative coupling of Stokes and Darcy flows. *SIAM J. Numer. Anal.*, 42(5):1959–1977, 2005.

[54] P.G. Saffman. On the boundary condition at the surface of a porous media. *Stud. Appl. Math., L*, (2):93–101, 1971.

[55] P. Song, C. Wang, and I. Yotov. Domain decomposition for Stokes-Darcy flows with curved interfaces. *Procedia Computer Science*, 18:1077–1086, 2013.

[56] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Internat. J. Comput. & Fluids*, 1(1):73–100, 1973.

[57] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer-Verlag Berlin Heidelberg, 2005.

[58] D. Vassilev, C. Wang, and I. Yotov. Domain decomposition for coupled Stokes and Darcy flows. *Comput. Methods Appl. Mech. Engrg.*, 268:264–283, 2014.

[59] D. Vassilev and I. Yotov. Coupling Stokes-Darcy flow with transport. *SIAM J. Sci. Comput.*, 31(5):3661–3684, 2009.

[60] M. F. Wheeler, G. Xue, and I. Yotov. A multiscale mortar multipoint flux mixed finite element method. *ESAIM: Mathematical Modelling and Numerical Analysis (M2AN)*, 46(4):759–796, 2012.

[61] M. F. Wheeler and I. Yotov. A posteriori error estimates for the mortar mixed finite element method. *SIAM J. Numer. Anal.*, 43(3):1021–1042, 2005.

[62] C. L. Winter and D. M. Tartakovsky. Groundwater flow in heterogeneous composite aquifers. *Water Resour. Res.*, 38(8), 2002.

[63] D. Xiu and G. E. Karniadakis. Modeling uncertainty in steady state diffusion problem via generalized polynomial chaos. *Comput. Methods Appl. Mech. Eng.*, 191:4927–4948, 2002.

[64] D. Zhang. *Stochastic Methods for Flow in Porous Media: Coping with Uncertainties*. Academic Press, San Diego, Calif., 2002.

[65] D. Zhang and Z. Lu. An efficient, high-order perturbation approach for flow in random porous media via Karhunen–Loève and polynomial expansions. *J. of Comp. Physics*, 194(2):773–794, 2004.